N° 73 mensuel Octobre 1991-24 F

La Revue de l'Utilisateur PC

Le magazine professionnel en programmation PC

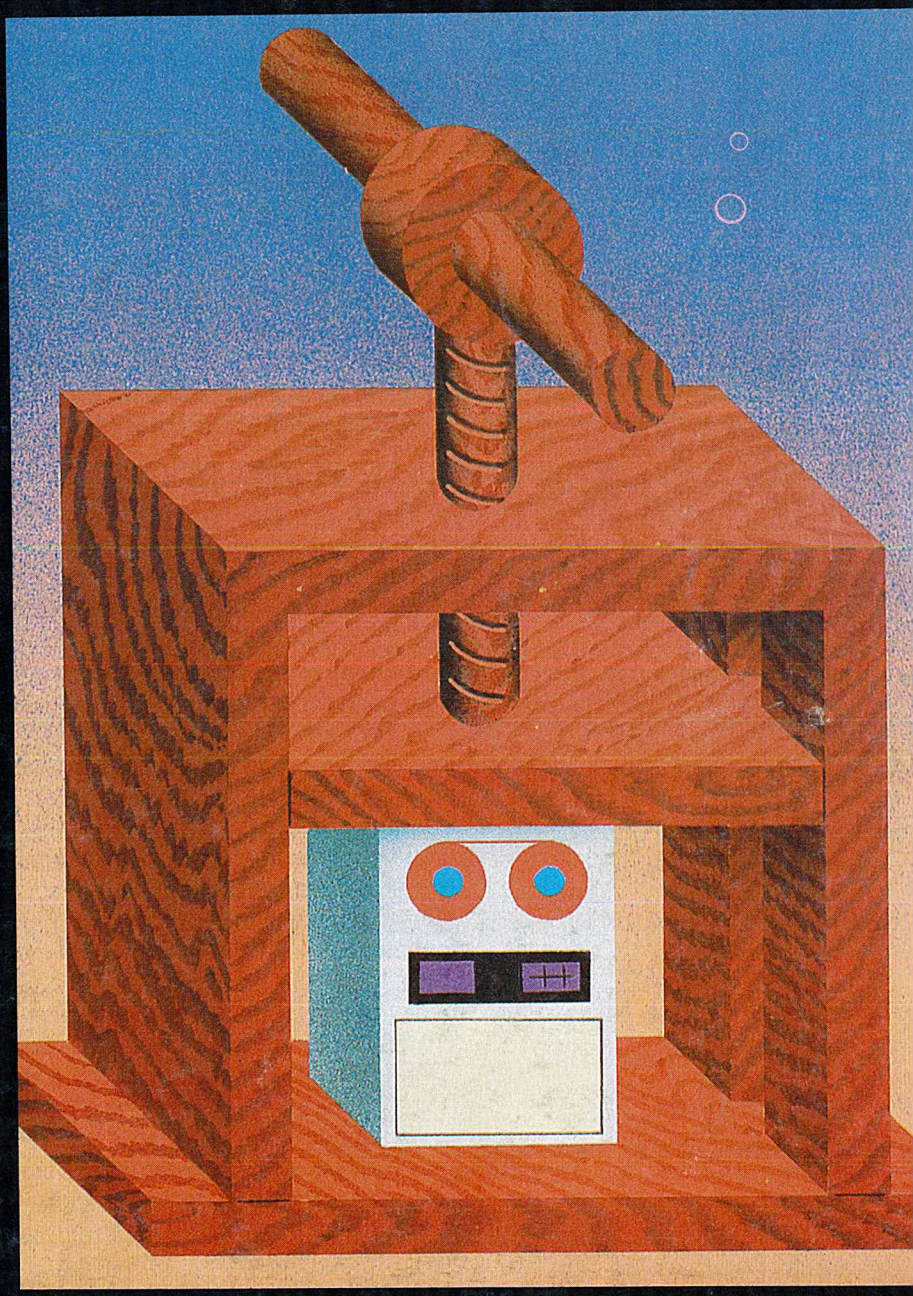
PROGRAMMATION ORIENTEE OBJET en ASSEMBLEUR : Une implémentation du Polymorphisme dynamique

UTILITAIRE : Le Prétexte (suite et fin)

VIRUS : Contrôle d'intégrité d'un disque dur

WINDOWS 3.0 et la mémoire

BRICOLINFO

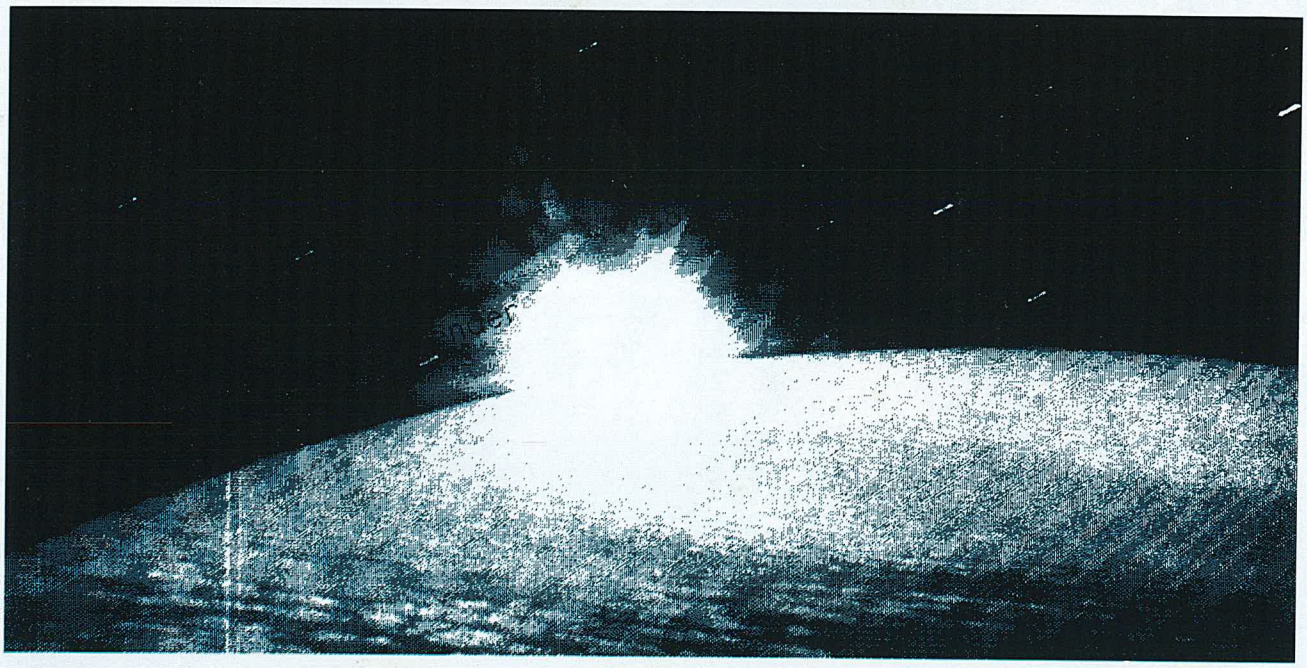


OCTOBRE 1991 - N° 73 - 24 F - Suisse 8 FS - Belgique 200 FB - Canada 6,25 $ Can.

M 2537 - 73 - 24,00 F

C AU-DELA DE LA LIMITE DE 640 Ko DU DOS

Nous vous ouvrons un nouveau monde de mémoire.



Laissez le nouveau 286/DOS-Extender™ de Phar Lap transformer votre compilateur Microsoft C en un outil puissant multi-mégaoctet !

Il n'a jamais été facile de programmer en C au-delà de la limite 640 Ko du DOS.

Vous avez beaucoup entendu parler dernièrement des DOS Extenders (DOS étendu) et de leur aptitude à vous laisser créer de gros programmes au-delà de la barrière de 640 Ko DOS. Aujourd'hui Phar Lap® rend cela plus facile que jamais ! Avec votre Microsoft C et notre nouveau 286/DOS-Extender, vous possédez tous les outils dont vous avez besoin pour créer rapidement et aisémemnt des applications en mode protégé multi-Mo - souvent rien qu'en réassemblant sans même effectuer des changements de code source. 286/DOS-Extender vous permet de construire des programmes ayant la place d'accueillir plus de caractéristiques et de fonctionnalités, sans avoir à souffrir avec des recouvrements ou des EMS. Mais la meilleure nouvelle est que vous n'avez besoin d'abandonner aucun de vos outils Microsoft C, y compris votre débogueur CodeView. Et avec 286/DOS-Extender, vous aurez non seulement beaucoup de mémoire pour vos applications, mais aussi pour le compilateur Microsoft C 6.0 lui-même. Plus de messages crispants sur la mémoire "hors du tas" ("out of heap") pendant la compilation sous Windows ou avec les réseaux.

Compatibilité totale

Etant donné que 286/DOS-Extender est intégré à votre programme, il est invisible à l'utilisateur final. Votre programme ressemble exactement à n'importe quelle autre application sous DOS. Il n'y a aucun environnement d'exploitation nouveau à acheter ou à apprendre pour l'utilisateur final. N'importe lequel des 30 millions de PC 80286, 386 ou 486 pouvant tourner sous MS-DOS ou PC-DOS peut fonctionner sous 286/DOS-Extender. Et comme les produits Phar Lap supportent les XMS, VCPI, et DPMI standards, les applications créées avec 286/DOS-Extender peuvent fonctionner dans des environnements variés, sous MS-DOS, y compris DeskView et tous les modes de Microsoft Windows. Prix: 4950 FF HT

Toutes les marques citées sont les marques déposées de leurs propriétaires respectifs.

Une technologie qui a fait ses preuves

Phar Lap est aussi le développeur du renommé 386/DOS-Extender™, qui a été utilisé dans plus de 600 applications comprenant AutoCAD 386, Mathematica DOS-386 et IBM Interleaf Publisher. 386/DOS-Extender est conçu pour les programmes qui nécessitent une optimisation de la vitesse et des résultats en 32 bits sur PC 386 ou 486. En utilisant une des techniques de notre DOS-Extender, les chefs d'industrie conservent leur part compétitive et délivrent toutes les fonctionnalités et les capacités exigées par leurs clients.

386/DOS Extender version 4.0

386/DOS Extender version 4.0 a toutes les caractéristiques de la version 3.0. Supporte DPMI ("DOS Protected Mode Interface") et MS Windows 3.0 en mode réel, standard et avancé (livrée avec la documentation de la version 3.0 et un nouveau manuel expliquant comment Windows et DPMI travaillent avec 386/DOS Extender). Supporte XMS, est compatible DOS 5.0, produit des fichiers .EXE au lieu de .EXP

Pour tous renseignements contacter:
SOFTWORLD
17, Av. Emile Zola, 75015 Paris.
Téléphone : (1) 40 59 02 99 / Fax : (1) 45 79 95 55

La Revue de l'Utilisateur PC

Le magazine professionnel en programmation PC

Mensuel 24 F
2, rue d'Amsterdam
75009 Paris
Tél. : (1) 42.81.54.27.
Service abonnement :
Tél. : (1) 42.81.54.27.

Directrice de la Publication :
Sardie ERRERA

Editeur :
Roger CHRISTOPHE

Rédacteur en chef :
Daniel VASSEUR

Directeur de fabrication :
Michel MILLET

Directeur de publicité :
Liliane ERRERA
Tél. : (1) 42.67.93.80.

Edité par SOFTin Communications (France) Sarl
R.C. Paris B 329 834 592
N° Siret
329 834 592 000 19
Commission Paritaire
(en cours)
Dépôt légal :
4e trimestre 1991

CCP Paris 8850 84 D

Impression :
SNIL
93600 Aulnay-sous-Bois

Composition :
SOFTin Communications

SOMMAIRE

2

PIC, IFF, IMG, PCX, TIFF... UNE EXPEDITION DANS LA JUNGLE DES FORMATS GRAPHIQUES 3e PARTIE : LE FORMAT IFF-LBM

Nous allons aujourd'hui étendre nos possibilités de conversion graphiques avec IFF, un format très répandu aussi bien à l'intérieur qu'à l'extérieur du monde PC
Par Jean-Michel SPINER

18

LE PRETEXTE (SUITE ET FIN)

Ce programme Prétexte dont la première partie a été publiée dans le n° 71 (Juin-Juillet 1991) de la Revue de l'Utilisateur PC permet de ménager la couche de phosphore de vos tubes cathodiques tout en distrayant votre esprit las et soucieux grâce à de multiples arabesques
Par Hugues FERRAND

32

PROGRAMMATION ORIENTEE OBJET EN ASSEMBLEUR : UNE IMPLEMENTATION DU POLYMORPHISME DYNAMIQUE

Cet article se propose de montrer comment on peut utiliser les techniques de la Programmation Orientée Objet (POO) en Langage Assembleur ; à la différence des autres langages qui ont été prévus pour cela, ou qui possèdent une surcouche objet, l'Assembleur ne possède pas d'outils spécifiques pour se servir des techniques de la POO
Par Paul FRANCESCHI

38

CONTRÔLE D'INTEGRITÉ D'UN DISQUE DUR

Les salles machines de notre Institut sont des lieux privilégiés pour la prolifération des virus
Par Patrick POLLET

49

WINDOWS 3.0 ET LA MÉMOIRE

Il n'y pas 1 mais 3 WINDOWS 3.0
Par Daniel VASSEUR

54

BRICOLINFO

Le dopage d'une machine ancienne n'est pas une mince affaire : nombreux sont les pièges à déjouer et les astuces à connaître
Par Daniel VASSEUR

PIC, IFF, IMG, PCX, TIFF... Une expédition dans la jungle des formats graphiques (3e Partie) Le format IFF-LBM

Nous allons aujourd'hui étendre nos possibilités de conversion graphiques avec IFF, un format très répandu aussi bien à l'intérieur qu'à l'extérieur du monde PC.

Jean-Michel SPINER

IFF (ou LBM) est exploité par ''Deluxe-Paint'', l'un des rares logiciels 2D micro à connaître un égal succès sur différents systèmes, notamment sur Amiga.

Nous allons donc disposer, grâce à sa connaissance, d'une corde de plus à notre arc pour les conversions sur PC, et d'une véritable passerelle avec le monde Amiga.

Même si le hard graphique PC s'est largement remis à niveau ces derniers temps, l'Amiga reste une référence en matière d'images, grâce aux superbes logiciels 2D et 3D qui ont été spécifiquement développés pour ce système.

Principes de l'IFF

D'une conception originale, IFF ou Interchange File Format n'est pas spécifiquement destiné aux données graphiques, mais peut aussi véhiculer du texte, de la musique, des fontes...

Dans un fichier IFF, toutes les informations sont regroupées dans des entités appelées des chunks.

Les chunks sont eux-mêmes les composantes d'une structure plus vaste appelée form.

Les forms

Une form comprend un en-tête de 12 octets :

— octets 1 à 4 : les quatre premiers octets contiennent le mot réservé FORM ;

— octets 5 à 8 : la taille de la form est codée sur 32 bits. Indépendamment du système ou du langage utilisé, on a un nombre entier positif égal à octet8 + 256*octet7 + 256*256*octet6 + 256*256*256*octet5.

— octets 9 à 12 : indicatif du type de form. Les IFF graphiques Amiga ou DeluxePaint PC auront à cet endroit la séquence ILBM, ce qui signifie Interleaved Bitmap.

Arrivée ici, la fonction de lecture s'attend à trouver un certain nombre de chunks dont la somme des tailles sera égale à la taille définie dans l'en-tête de la form.

Les chunks

Un chunk possède un en-tête de 8 octets :

— octets 1 à 4 : type de chunk,

— octets 5 à 8 : taille du chunk sur 32 bits.

Un fichier graphique de type ILBM se doit de posséder au moins les trois chunks suivants :

— BMDH : Bitmap Header,
— CMAP : Color Man,
— BODY : Données.

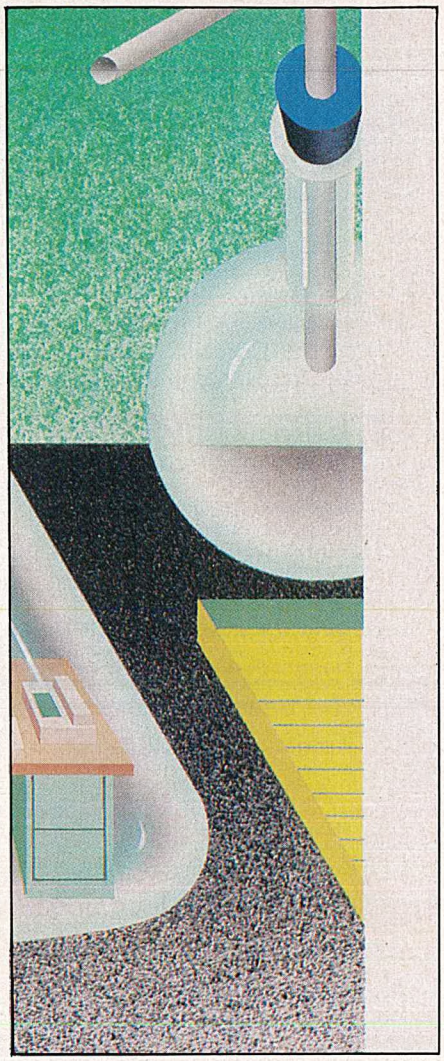
Chaque logiciel peut, pour des besoins spécifiques, créer ses propres chunks, qui pourront par exemple fournir des indications d'animation ou de déformation de l'image.

Dans la mesure où ces chunks possèdent un en-tête standard, les données qu'ils contiennent seront ignorées par un logiciel inapte à les exploiter, et la fonction de lecture passera directement au chunk suivant.

L'ordre dans lequel apparaissent les chunks est théoriquement quelconque. Dans la pratique, le bitmap header précède la table de couleur, et les datas sont reléguées en fin de fichier.

Un algorithme général de lecture peut ainsi être dressé :

```
tant que fin de form non atteinte
   Lire identification chunk
   Lire taille chunk
   Si chunk connu
      Traiter
   Sinon
      Avancer de taille dans fichier
fin tant que
```



Bitmap header

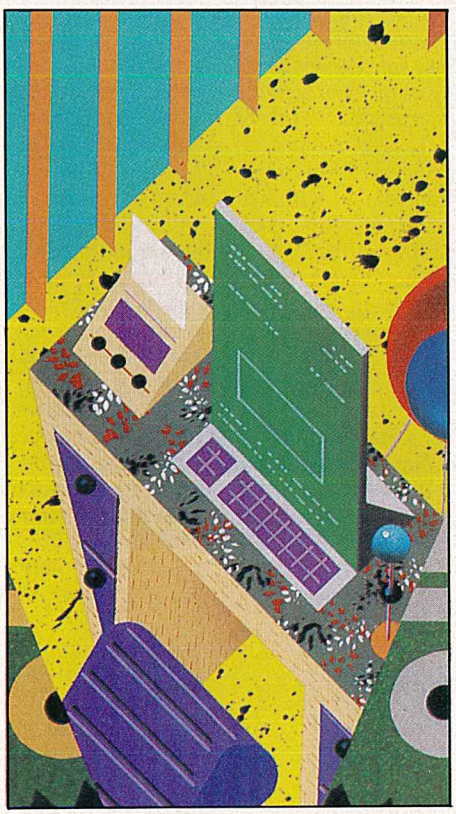
En général, le chunk BMDH est formé de vingt octets :

— longueur de l'image (2 octets),
— hauteur de l'image (2 octets),
— origine x de l'image (2 octets),
— origine y de l'image (2 octets),
— nombre de plans mémoire (1 octet),
— bidon (1 octet),
— compression (2 octets),
— bidon (2 octets),
— rapport longueur/hauteur d'un pixel (2 octets),
— résolution en x (2 octets),
— résolution en y (2 octets).

Les valeurs x,y de l'origine de l'image permettent de manipuler des images dont la taille est différente de la résolution de l'écran.

Le nombre de plans mémoire est égal au nombre de bits nécessaires pour coder un pixel : 1 plan pour une image monochrome, 4 plans pour 16 couleurs, 8 plans pour 256 couleurs.

Les deux octets de compression sont à 1 si l'image suit un codage RLC, à 0 dans le cas contraire.



Color map

Le chunk color map renferme la précieuse table de couleurs. Rappelons-en brièvement le principe.

Une teinte est définie par ses valeurs dans les trois couleurs élémentaires, rouge, vert, et bleu. Le dispositif hardware VGA nous permet de choisir, pour chaque composante, 64 valeurs différentes.

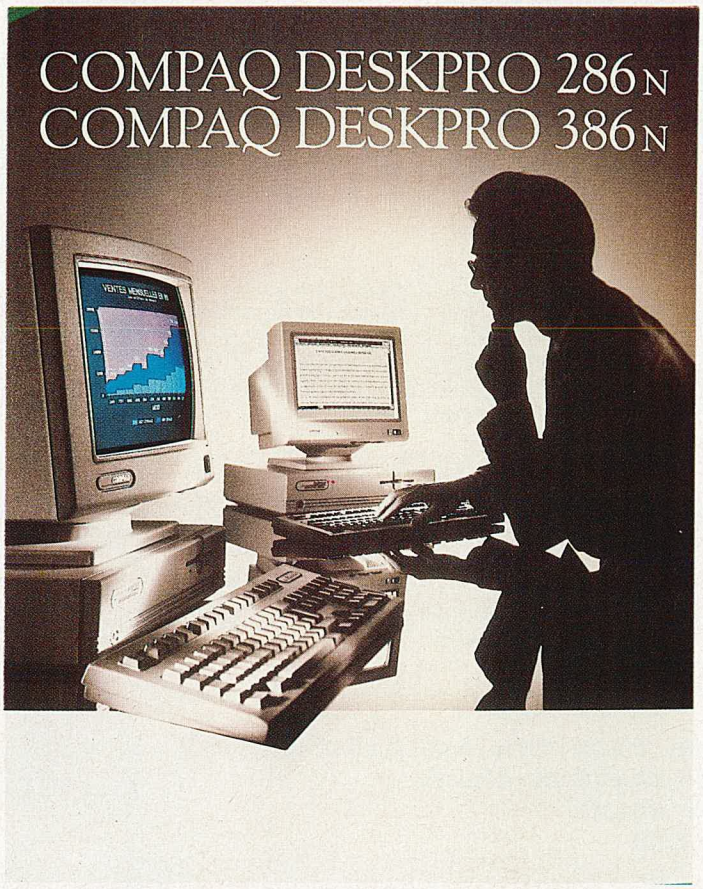
Le nombre total de teintes disponibles est donc de 64*64*64 = 262144, valeur que l'on doit coder sur au moins 18 bits.

Or, ce même dispositif hardware ne nous autorise que 8 bits pour coder la couleur d'un pixel en mémoire vidéo.

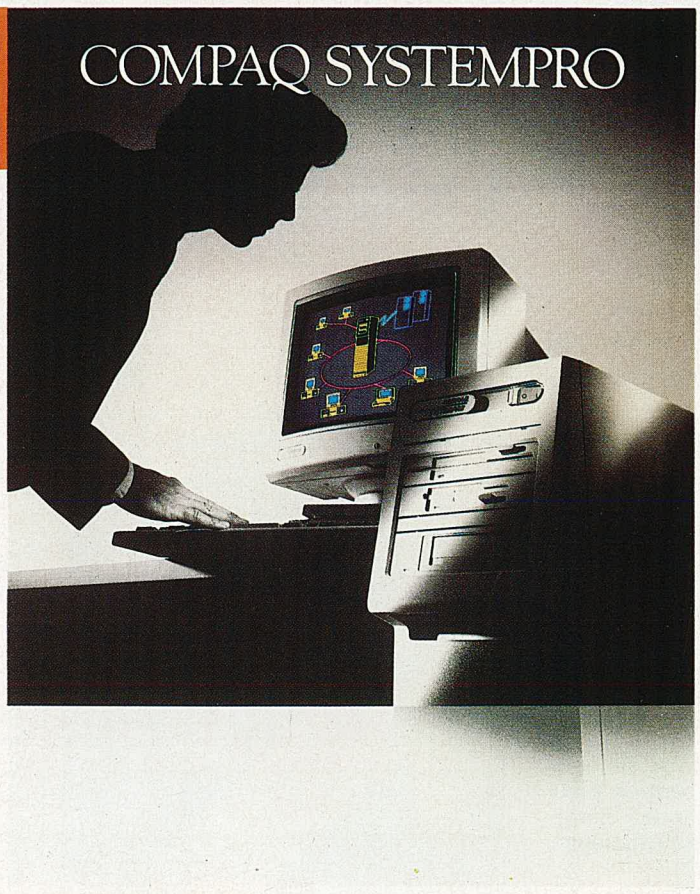
C'est pourquoi chaque pixel se verra affecté, non une valeur brute de couleur, mais un numéro d'entrée dans la table de couleur, qui elle, contient la véritable correspondance avec la teinte désirée.

La conséquence la plus funeste de cette contrainte est la restriction à 256 du nombre de teintes affichées simultanément : en mode VGA 13h par exemple, on dispose de 256 couleurs simultanées parmi 262144.

Equipés respectivement des microprocesseurs 286 à 12 MHz et 386SX à 16 MHz, le **COMPAQ DESKPRO 286N** et le **COMPAQ DESKPRO 386N** offrent en standard toutes les fonctionnalités des micro-ordinateurs professionnels : 3 connecteurs d'extension (2 connecteurs 8/16 bits ISA et 1 connecteur dédié à la mémoire vive rapide), 1 Mo de mémoire vive extensible à 13 Mo (286N) et 16 Mo (386N), jusqu'à 120 Mo de mémoire de masse en interne, 2 emplacements de mémoire de masse, un nouveau contrôleur graphique VGA accéléré avec support du mode texte 132 colonnes.

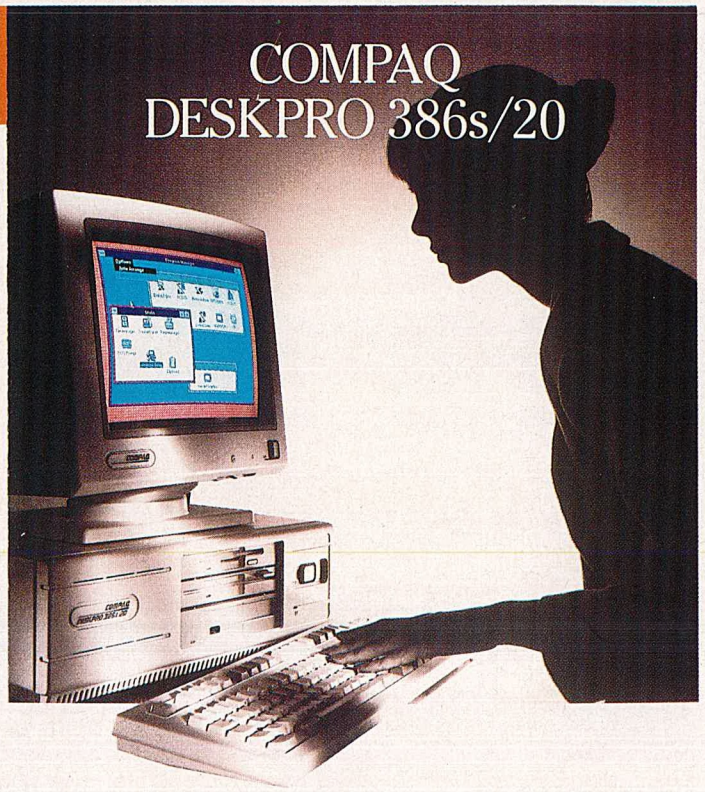


COMPAQ DESKPRO 286 N
COMPAQ DESKPRO 386 N



Basé sur une architecture évolutive en termes de processeur et de mémoire de masse, le **COMPAQ SYSTEMPRO** a été le premier micro-ordinateur à offrir un mode multi-processeur et un système multidisque. Il évolue ainsi avec les besoins des utilisateurs et préserve leurs investissements. Cette nouvelle génération d'ordinateurs a connu un grand succès : de nombreuses entreprises sont en train d'adopter le **COMPAQ SYSTEMPRO** pour faire évoluer leurs sites centraux et leur mini-informatique vers une informatique ouverte et au standard de l'industrie.

COMPAQ SYSTEMPRO

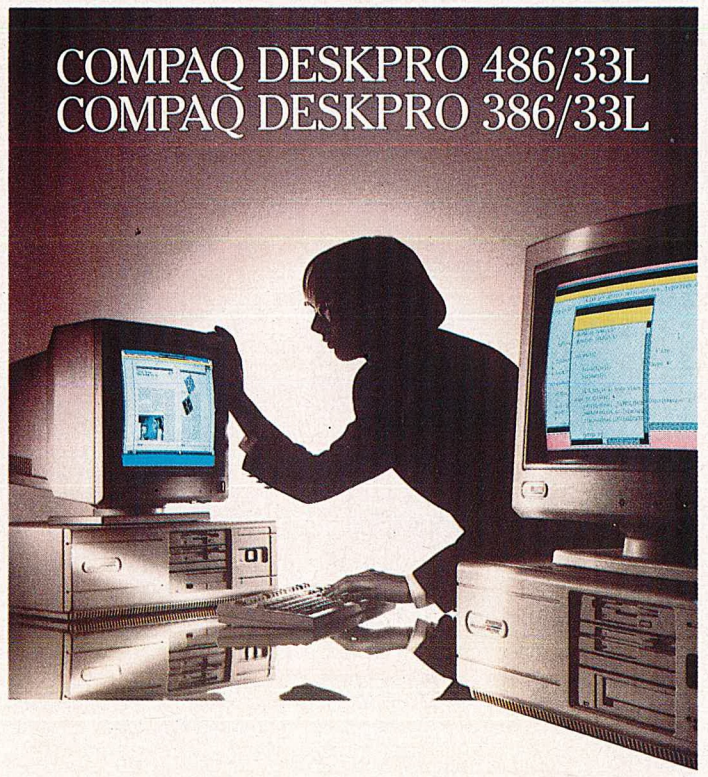


Equipé du dernier microprocesseur INTEL 80386SX à 20 MHz, le **COMPAQ DESKPRO 386s/20** est disponible en 3 modèles : 1,60 et 120 (disques durs avec temps d'accès inférieur à 19 ms). Outre sa mémoire cache ultra-rapide, le **COMPAQ DESKPRO 386s/20** est doté de nombreuses innovations. Son contrôleur VGA est ainsi accéléré pour offrir un maximum de performances en environnement graphique. De plus, il supporte le mode texte 132 colonnes utilisé par un nombre grandissant de logiciels de type tableur, traitement de texte ou d'émulation terminal.

COMPAQ DESKPRO 386s/20

Le **COMPAQ DESKPRO 486/33L** est aujourd'hui le micro-ordinateur de bureau le plus puissant du monde. En plus d'un coprocesseur 80387 et de 8 Ko de mémoire cache intégrés au processeur i486, sa carte processeur comprend un cache de deuxième niveau de 128 Ko de mémoire ultra-rapide. Ce nouveau micro-ordinateur est conçu pour des applications spécialisées techniques telles que la CAO, le développement logiciel, les SGBD et les applications scientifiques.

Le **COMPAQ DESKPRO 386/33L** dispose d'une carte processeur qui est équipée du microprocesseur 386 à 33 MHz et de 64 Ko de mémoire cache. Il constitue l'outil idéal pour les applications spécialisées telles que la CAO, les SGBD, la modélisation et l'analyse financière, ainsi que la PAO.
Un programme d'échange de la carte processeur 386 contre la carte i486 va permettre d'assurer aux utilisateurs la pérennité de leurs investissements, tout en répondant à leurs besoins d'évolutivité.



LA DEDICACE INFORMATIQUE

COMPAQ
REVENDEUR AGREE

173, rue de Vaugirard 75015 PARIS
Tél. 45.66.94.69
Fax : 43.06.35.18 - Télex 202 993

COMPAQ DESKPRO 486/33L
COMPAQ DESKPRO 386/33L

Création Parenthèse - Paris - 43 55 76 20

Pour la commodité de lecture, les couleurs de la color map ne sont pas sur 18 bits mais sur 24 bits, un octet par composante.

La lecture et l'affectation aux registres vidéo se font de la façon suivante.

```
struct coul {
   unsigned char r,v,b;
}couleur[256];
int i1,nb_couls;
long longueur;
FILE *source;

 ...
fread(couleur,(int)longueur,1,source);
for (i1=0;i1<nb_couls;i1++) {
   Set_Palette_Color
   (i1,couleur[i1].r,couleur[i1].v,couleur[i1].b);
}
 ...
```

Data

Dans notre programme exemple, OUTIFF, nous n'allons traiter que les images au format compressé. Pour les autres, les données figurent séquentiellement dans le chunk body, ligne par ligne, et plan par plan au sein de chaque ligne.

Par exemple, une ligne de pixels de couleurs 1 sera composée de 80 octets FF en mode CGA, de 320 octets 01 en VGA 256 couleurs, et de 4 fois 80 octets (3 fois 00 et une fois FF) en VGA 16 couleurs.

Le décodage en cas de compression est un peu plus lent et compliqué, ce qui est compensé par le gain d'espace disque.

En codage RLC, le taux de compression est d'autant plus favorable que l'image est simple, et comporte de nombreuses zones de couleur identique.

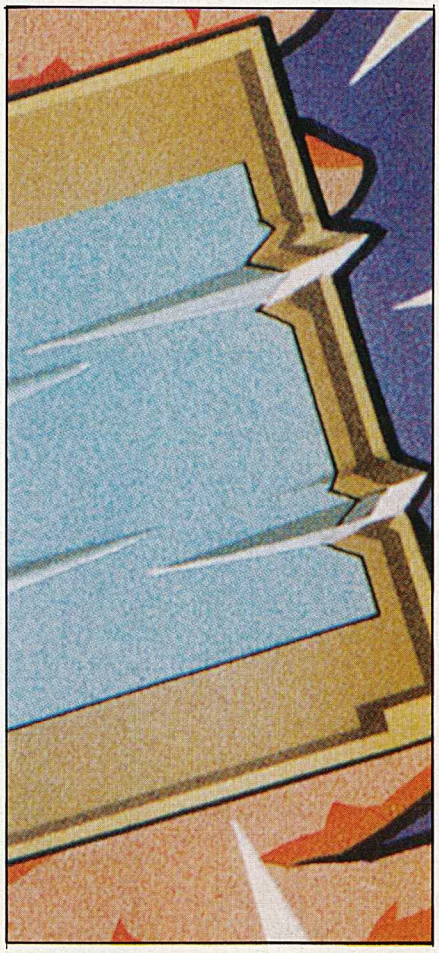
Il est en effet plus concis de coder l'information "50 pixels identiques" et "couleur verte" que 50 fois l'information "couleur verte".

Les règles de désencodage est la suivante : les octets sont lus séquentiellement. Si la valeur lue est supérieure à 128, on aura (257 – valeur) fois la répétition de la valeur de l'octet suivant.

En revanche, si la valeur est inférieure à 128, on aura (valeur + 1) octets différents à suivre. Enfin, si la valeur est égale à 128, elle sera simplement ignorée.

L'algorithme de lecture est le suivant :

```
Tant que non fin de chunk body
   lire carlu
   si carlu > 128
      nbr_bytes = 257 - carlu
      val = carlu suivant
   sinon, si carlu < 128
      nbr_bytes = carlu + 1
      de 1 à nbrbytes
         val = carlu suivant
```



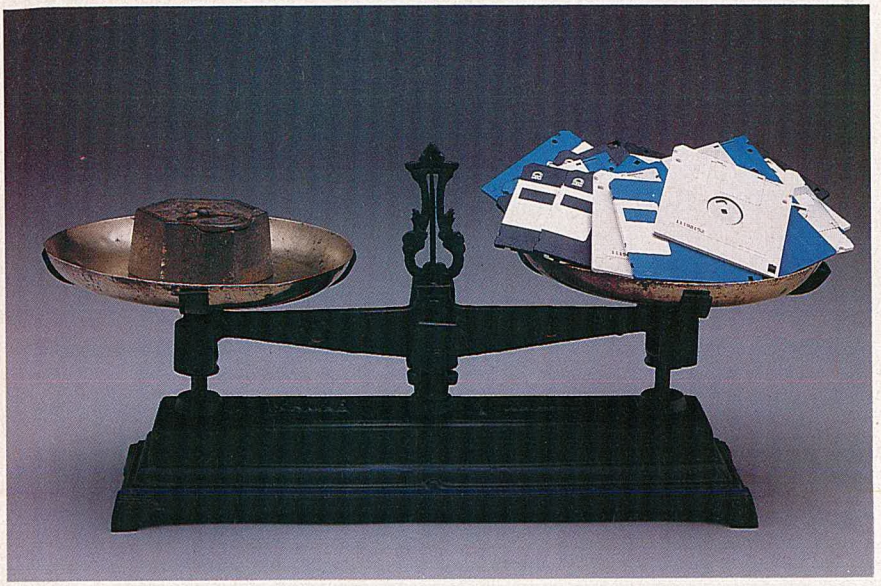
Le programme OUTIFF

L'exemple de ce mois-ci, OUTIFF, repose sur le même principe que l'OUTPCX du mois dernier.

OUTIFF est capable d'afficher une image ou partie d'image IFF ayant été sauvée dans l'un des modes suivants :

```
- CGA      640*200    06H
- EGA      640*350    10H
- VGA1     640*480    12H
- VGA2     320*200    13H
- VGA3     800*600    29H
```

Disquettes 3" 1/2 720 : 155 F le kilo *



DISQUETTES NEUTRES

garanties 100 % sans erreurs livrées
avec étiquettes, sticker et pochettes.
Prix unitaires TTC, TVA 18,60 % incluse.

PAR QUANTITE DE 10 :

Disquette 3" 1/2 1,44 Mo	8,70 F
Disquette 5" 1/4 360 Ko	1,85 F
Disquette 5" 1/4 1,2 Mo	4,40 F

PAR QUANTITE DE 100 :

Disquette 3" 1/2 1,44 Mo	7,50 F
Disquette 5" 1/4 360 Ko	1,80 F
Disquette 5" 1/4 1,2 Mo	4,20 F

PAR QUANTITE DE 600 :

Disquette 3" 1/2 1,44 Mo	6,80 F
Disquette 5" 1/4 360 Ko	1,75 F
Disquette 5" 1/4 1,2 Mo	3,90 F

NOUVEAU

FAX cleaner : **120 F** TTC
Laser PRINT cleaner : **120 F** TTC.

SCANNER COULEUR PC :
256 coul. : **3700 F** TTC

SCANNER NOIR / BLANC
avec PC PAINT BRUSH +
1290 F TTC

+ TIMEWORKS PUBLISHER
2360 F TTC

Pour les COLLECTIONNEURS

T.SHIRT + PIN'S	85,00 F TTC
(participation frais d'envoi)	15,00 F
PIN'S seul	35,00 F TTC
(participation frais d'envoi)	15,00 F

T. SHIRT SOURIS

Taille mode	65,00 TTC
(participation frais d'envoi)	15,00 F

CATALOGUE COMPLET SUR DEMANDE

DIVERS CONSOMMABLES

Tapis pour souris	30 F
Support imprimante	150 F
Pochette étiquettes (5" 1 4)	50 F
Pochette étiquettes (3" 1 2)	50 F
Rangement tiroir 3" 1 2 et cartouche numériques	190 F
Rangement tiroir 5" 1 4 et disque laser	249 F

TRACKBALL : 395 FTTC

BOITES DE RANGEMENT

5" 1/4 par 100	75 F
5" 1/4 par 50	55 F
5" 1/4 par 10	30 F
3" 1/2 par 80	75 F
3" 1/2 par 40	55 F
3" 1/2 par 10	30 F
MICRO 15 3" 1/2	39 F

SOURIS OPTIQUE PC : 695 F TTC
résolution 100 à 800 DPI, 3 vitesses.

Disquettes de nettoyage
(3" 1/2 ou 5" 1/4) : 70 F TTC

SOURIS MECANIQUE :

AM 5 000 PC	210 F TTC
AM 5 000 Plus PC avec logiciel dessin	290 F TTC
Mousse ATARI	270 F TTC
Souris Opto-mécanique bi-compatible (ATARI / AMIGA)	199 F TTC
Souris optique ATARI / AMIGA	495 F TTC
Souris optique IBM	695 F TTC
Trackball	395 F TTC

Disquette 3" 1/2 720 Ko en boîte plastique par 10 :

l'unité x 10	4,50 F
l'unité x 100	4,30 F
l'unité x 600	4,10 F

Disquette 3" 1/2 720 Ko en boîte plastique par 20 :

la boite de 20	98,00 F
par 100 disquettes (la boite)	95,00 F
par 600 disquettes (la boite)	93,00 F

*** 1 kilo = 50 disquettes**

Modèle	Qté	Prix unitaire TTC	Prix total TTC
Forfait port et emballage minimum.			50 F
		Total TTC :	

BON DE COMMANDE

A retourner à :
PC/S 18 - 5, rue J.-F. LEPINE - 75018 Paris
Tél. : (1) 42 05 96 66 - Fax : (1) 46 07 97 60

Je vous prie de bien vouloir noter ma commande référencée ci-contre.

Nom : .....
Société : .....
Adresse : .....
Code postal : .....
Ville : .....
Je désire régler par : ☐ Chèque
☐ Carte bleu N° .....
☐ Carte American Express n° ..... Signature
Durée de val. : .....
Date : .....

Toutes les marques citées sont déposées.
Prix modifiables sans préavis.
Possibilité de crédit après étude du dossier. Crédit étudiant.
Au dessus de 200 disquettes et pour tout envoi au dessus de 15 kg, nous consulter.

Ce programme, a été rédigé en langage C et en assembleur. Voici le listing du fichier make.

```
#
#         MAKE outiff
#

.c.obj:
   cl -c -W1 -Ze -AS -G2 $*.c

.asm.obj:
   masm /DSMALL $*.asm,$*.obj,nul.lst,nul.crf;


outiff.obj : outiff.c

putpixel.obj : putpixel.asm

outiff.exe : outiff outiff.obj putpixel.obj
link /ST:5000 outiff+putpixel,,,;
```

L'utilisation du module assembleur est facultative. En effet, OUTIFF possède deux modes d'affichage : BIOS ou écriture directe en mémoire d'écran.

Seules les routines autorisant la deuxième possibilité sont en assembleur.

On pourra donc, sous-réserve de mettre en remarque les appels à ces fonctions, ne pas linker OUTIFF.OBJ avec PUTPIXEL.OBJ.

Le recours au BIOS, s'il garantit un parfait fonctionnement avec types de configuration, allonge le temps d'affichage de façon désolante.

Syntaxe d'OUTIFF :

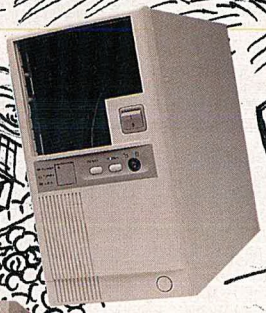
outiff [-l] [-c] [-b] [-p] [-k] nom[.lbm]

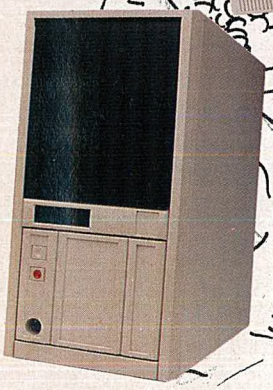
Les paramètres sont destinés à faciliter l'utilisation d'OUTIFF dans des fichiers BATCH, et permettent ainsi d'avoir des fichiers de commande de type scénario.

On pourra aisément de cette façon créer des demos, et des boucles d'affichage interactives.

Construisez... sur du solide !

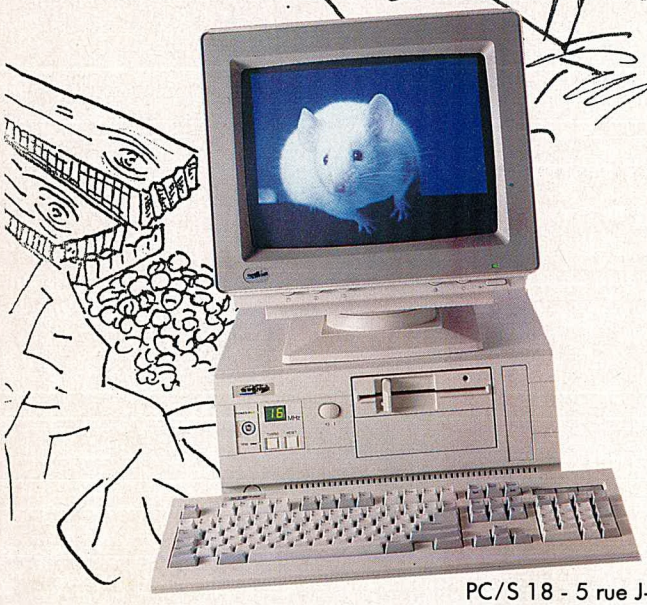
CHANTIER OUVERT
AU PUBLIC
de 10 H à 19 H
du lundi au samedi





AT 386 SX - 16 1Mo
Boitier Baby - 1 Mo de RAM - Disque Dur 40 Mo. VGA Couleur
8 800,00 F TTC
Avec Boitier Baby Plus ou Mini Tour : **9 300,00 F TTC**
AT 386 SX - 20 2 Mo
Lecteurs 3" 1/2 et 5" 1/4
10 400,00 F TTC

AT 386 - 25 2 Mo
Disque Dur 89 Mo - VGA Couleur 1024 x 768 - Dos 4.01....
14 800,00 F TTC
Avec 64 Ko Mémoire cache : **16 200,00 F TTC**
AT 386 - 33 4 Mo extensible à 16
Disque Dur 128 Mo
18 500,00 F TTC



AT 286-12 1Mo
Disque Dur 20 Mo - Monochrome
4 650,00 F TTC

AT 286-16 1 Mo
Disque Dur 40 Mo. ATBUS - VGA Monochrome
5 450,00 F TTC

Maintenance assurée par ALCATEL DSF sur toute la France.

PC/S 18 - 5 rue J-F Lépine, 75018 PARIS **Tél. : (1) 42 05 96 66** - Fax : (1) 46 07 97 60

Signification des paramètres

[-l] : le programme se termine avec un code de sortie égal au code ascii d'un caractère frappé. On peut ainsi agir sur le déroulement des séquences et proposer des images qui font office de menus.

[-c] : sortie immédiate du programme avec un code de 0 si aucune touche n'a été frappée, et de EXIT_KEYPRESSED (= 101) dans le cas inverse. Par défaut, la sortie est provoquée par l'appui sur une touche et le code retour est 0.

[-b] : utilisation du BIOS. Cette option peut se révéler utile avec certains types de cartes écran.

[-p] : il n'est pas tenu compte de la palette de l'image. Outre l'obtention d'effets saisissants, cette possibilité permet d'afficher plusieurs images possédant la même palette sans avoir à réinitialiser celle-ci. (Environ 4 secondes en mode VGA 256 couleurs).

[-k] : keep graph mode. A la sortie d'OUTIFF, le mode graphique est conservé. Cela permet d'éviter les effets de flash, et de conserver éventuellement la palette. Il est à noter que le mode graphique adéquat n'est activé en début de programme, que dans la mesure où il ne s'agit pas du mode courant.

Voici un exemple de fichier de commande :

test.bat permet de choisir une séquence à visionner (1) ou (2), de la voir en boucle jusqu'à la frappe sur une touche, puis de retourner à l'image menu, jusqu'à ce que l'option de sortie (3) soit activée.

Les valeurs de sortie de l'affichage de menu.lbm sont respectivement égales aux codes ascii des caractères 1, 2, et 3, soit 49, 50, et 51.

```
@echo off
:debut
outiff -1 -k menu
if errorlevel 51 goto fin
if errorlevel 50 goto seq2
if errorlevel 49 goto seq1
goto debut
:seq2
outiff -c -k seq2_01
if errorlevel 101 goto debut
:boucle2
outiff -c -p -k seq2_02
if errorlevel 101 goto debut
outiff -c -p -k seq2_01
if errorlevel 101 goto debut
goto boucle2
:seq1
outiff -c -k seq1_01
if errorlevel 101 goto debut
:boucle1
outiff -c -p -k seq1_02
if errorlevel 101 goto debut
outiff -c -p -k seq1_01
if errorlevel 101 goto debut
goto boucle1
:fin
mode 80
```

Pour chaque séquence, on commence par afficher la première image afin d'initialiser la palette, puis on boucle sur l'affichage des images avec l'option -p (non prise en compte de la palette).

La frappe d'une touche provoque le code de sortie EXIT_KEYPRESSED, ce qui renvoit à l'affichage du menu.

Bulletin d'abonnement pour 1 an

10 numéros au prix de : France : 220 F T.T.C., Etranger : 300 F
Je souscris un abonnement d'un an à la Revue de L'utilisateur de l'IBM-PC

Nom : .....
Société : .....
Adresse : .....
.....
Code postal : ..... Ville : ..... Tél. : .....

Chèque bancaire joint à l'ordre de SOFtin Communications (France).

A renvoyer à : La Revue de l'Utilisateur de l'IBM—PC, 2, rue d'Amsterdam, 75009 Paris.

A ce prix-là ça va faire des jaloux !!!

AS 80386SX-20 MHz

1 Mo de RAM Ext. 8 Mo — Bios AMI — 6 slots Libre — 2 ports séries, 1 port // — Support copro. 80387SX — Alimentation 200 watts — 1 lecteur 3''1/2 1.44 Mo — Boîtier Baby ou Desk — Clavier 102 touches — MS-DOS 4.01 — **Garantie 1 an sur site**

Écran \ Disque Dur	40 Mo AT Bus	85 Mo AT Bus	125 Mo AT Bus	210 Mo AT Bus
Hercule	7562 F	8812 F	9375 F	11750 F
VGA Mono	8000 F	9250 F	9812 F	12187 F
VGA Couleur 800×600	9312 F	10562 F	11125 F	13500 F
VGA Couleur 1024×768	9625 F	10875 F	11437 F	13812 F

AS 80386-25 MHz

2 Mo de RAM Ext. 8 Mo — Bios AMI — 6 slots Libre — 2 ports séries, 1 port // — Support copro. 80387 — Alimentation 200/230 watts — 1 lecteur 3''1/2 1.44 Mo — Boîtier Baby ou Desk — Clavier 102 touches — MS-DOS 4.01 — **Garantie 1 an sur site**

Écran \ Disque Dur	40 Mo AT Bus	85 Mo AT Bus	125 Mo AT Bus	210 Mo AT Bus
Hercule	9687 F	10937 F	11500 F	13875 F
VGA Mono	10125 F	11375 F	11937 F	14312 F
VGA Couleur 800×600	11437 F	12687 F	13250 F	15625 F
VGA Couleur 1024×768	11750 F	13000 F	13562 F	15937 F

AS 80386-33 MHz

2 Mo de RAM Ext. 8 Mo — Bios AWARD — 64 Ko cache — 6 slots Libre — 2 ports séries, 1 port // — Support copro. 80387 — Alimentation 200/230 watts — 1 lecteur 3''1/2 1.44 Mo — 1 lecteur 5''1/4 1.2 Mo — Boîtier Baby ou Desk — Clavier 102 touches — MS-DOS 4.01 — **Garantie 1 an sur site**

Écran \ Disque Dur	40 Mo AT Bus	85 Mo AT Bus	125 Mo AT Bus	210 Mo AT Bus
Hercule	11750 F	13000 F	13562 F	15937 F
VGA Mono	12187 F	13437 F	14000 F	16375 F
VGA Couleur 800×600	13500 F	14750 F	15312 F	17687 F
VGA Couleur 1024×768	13812 F	15062 F	15562 F	18000 F

AS 80486-25 MHz

4 Mo de RAM Ext. 16 Mo — Bios AWARD — 6 slots Libre — 2 ports séries, 1 port // — Support copro. 80487 — Alimentation 200/230 watts — 1 lecteur 3''1/2 1.44 Mo — 1 lecteur 5''1/4 1.2 Mo — Boîtier Baby ou Desk — Clavier 102 touches — MS-DOS 4.01 — **Garantie 1 an sur site**

Écran \ Disque Dur	40 Mo AT Bus	85 Mo AT Bus	125 Mo AT Bus	210 Mo AT Bus
Hercule	19125 F	20375 F	20937 F	23312 F
VGA Mono	19562 F	20812 F	21375 F	23750 F
VGA Couleur 800×600	20875 F	22125 F	22687 F	25062 F
VGA Couleur 1024×768	21187 F	22437 F	22937 F	25375 F

MÉMOIRES

pour TOSHIBA

2 Mo T1200XE	2200 F
2 Mo T3200SX	2200 F
2 Mo T5200	2000 F

pour COMPAQ

2 Mo 386N	2700 F
2 Mo 386s/20	1800 F

pour Hewlett Packard

1 Mo HPIII, IIP	850 F
2 Mo HPIII, IIP	1400 F
4 Mo HPIII, IIP	2300 F

ONDULEURS

SHANDY Sinusoïdaux Sortie RS 232

500 VA	5850 F
750 VA	8900 F
1000 VA	11000 F
1250 VA	12300 F
2000 VA	19900 F
2500 VA	23900 F

IMPRIMANTES

STAR LC 24-15

Prix Public : 5480 F — NOTRE PRIX : 3600 F

24 aiguilles — 136 colonnes — 200 CPS — 14 jeux de 14 caractères chacun — Styles d'écritures : Courrier, Prestige, Orator, Script — **Garantie 1 an retour atelier**

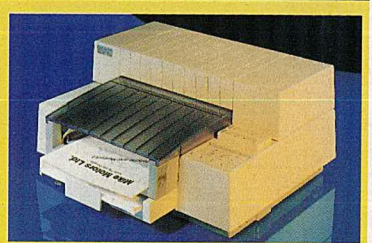
STAR LASER LP4

Prix Public : 11980 F — NOTRE PRIX : 7700 F

Version Postscript :

Prix Public : 16980 F — NOTRE PRIX : 10900 F

4 pages minute — 1 Mo RAM — Émulation : HP Laserjet II P — Epson FX850 — **Garantie 1 an sur site**



HEWLETT PACKARD DESKJET 500

Prix Public : 5990 F — NOTRE PRIX : 4300 F

Impression jet d'encre — Mémoire tampon — 16 Ko - **Garantie 3 ans et 60000 pages sans panne**

Frais de port non inclus.
Tous nos prix s'entendent hors taxes, T.V.A. 18,60 % en sus.

ACSE

5, rue des Lyonnais - 75005 PARIS
Tél. 16 (1) 43 37 05 06
Fax 16 (1) 43 37 15 46

SETRI

11, bd du Gal.-de-Gaulle - 92700 COLOMBES
Tél. (1) 47 81 42 56 +
Fax (1) 42 42 96 42

```
/*

      +-------------------------------------------+--------+
      !                 OUTIFF                    !  1.00  !
      +-------------------------------------------+--------+
      !                                                    !
      !     ± FONCTION   :   Affichage du format IFF       !
      !                      dans les rsolutions :         !
      !                      - 640*200   2 couleurs        !
      !                      - 640*350  16 couleurs        !
      !                      - 640*480  16 couleurs        !
      !                      - 320*200 256 couleurs        !
      !                      - 800*600  16 couleurs        !
      !                                                    !
      !                                                    !
      !     ± AUTEUR     :   Jean-Michel SPINER           !
      !                                                    !
      +----------------------------------------------------+

*/

#include <stdio.h>
#include <dos.h>
#include <malloc.h>
#include <string.h>



#define TRUE 1
#define FALSE 0
#define INT_VIDEO 0x10
#define CGA  0x06  /* CGA Mode  6H (640x200x2)    */
#define EGA  0x10  /* EGA Mode 10H (640x350x16)   */
#define VGA1 0x12  /* VGA Mode 12H (640x480x16)   */
#define VGA2 0x13  /* VGA Mode 13H (320x200x256)  */
#define VGA3 0x29  /* VGA Mode 29H (800x600x16)   */
#define TAILLE_BUFFER 16000

#define EXIT_OK 0
#define EXIT_KEYPRESSED 101
#define EXIT_ERR_SYNTAX 102
#define EXIT_FILE_NOT_FOUND 103
#define EXIT_ERR_HEADER 104
#define EXIT_UNKNOW_FORMAT 105
#define EXIT_ERR_ALLOC 106
#define EXIT_ERR_SETGRAPH 107

#define IFF_EXT ".LBM"

unsigned char buffer[TAILLE_BUFFER];
unsigned short graph_mode;
unsigned short txt_mode;


struct coul {
   unsigned char r,v,b;
}couleur[256];


unsigned char bmhd_lu = FALSE;
unsigned char body_lu = FALSE;
unsigned char str5 [5];
unsigned char strlong [5];
unsigned char str10K[10240];
char infile[80],outfile[80];
unsigned int w,h;
```



```
unsigned int xo,yo;
unsigned int compression;
unsigned char nb_plans;
long longueur;
FILE  *source,*cible;
unsigned char huge *data_zone;

/* PARAMETRES */
unsigned char wait_key = TRUE;
unsigned char use_bios = FALSE;
unsigned char keep_graph_mode = FALSE;
unsigned char use_default_palette = FALSE;
unsigned char exit_keypressed = FALSE;
unsigned char debug = FALSE;


long str_to_long (unsigned char *chaine);
unsigned int str_to_int (unsigned char *chaine);


/*   ------------------------------------------------   */
/*                          MAIN                        */
/*   ------------------------------------------------   */

main(argc,argv)
int argc;
char *argv[];
{

   register unsigned int i1;
   register unsigned int i2;
   int code_sortie;

   /* Rcup paramtre */
   if (argc<2) {
      fprintf (stderr,"\nSyntaxe : OUTIFF [-c] [-b] [-k] [-1] fichier[.LBM]\n\n")
;
      exit (EXIT_ERR_SYNTAX);
   }
   for (i1=1;i1<argc-1;i1++) {
      if (argv[i1][0]=='-') {
         switch (argv[i1][1]) {
            case 'b' : use_bios = TRUE; break;
            case 'c' : wait_key = FALSE; break;
            case 'd' : debug = TRUE; break; /* secret */
            case 'k' : keep_graph_mode = TRUE; break;
            case 'p' : use_default_palette = TRUE; break;
            case '1' : exit_keypressed = TRUE; break;
            default :
               fprintf (stderr,"\nErreur param : OUTIFF ",
               "[-c] [-b] [-k] [-1] fichier[.LBM]\n\n");
               exit (EXIT_ERR_SYNTAX);
               break;
         }
      }
   }

   strcpy (infile,argv[argc-1]);

   /* Existence fichier */
   if (strstr(strupr(infile),IFF_EXT)==NULL) strcat(infile,IFF_EXT);
   if ((source=fopen(infile,"rb"))==NULL) {
      fprintf (stderr,"\nFichier %s non trouve\n\n",infile);
      exit (EXIT_FILE_NOT_FOUND);
   }

```

```
09/17/1991  09:47        FICHIER  :  OUTIFF.C                    PAGE   3

   Lire_entete ();

   while ((fread(str5,4,1,source)==1)) {
      memset (strlong,0,5);
      fread(strlong,4,1,source);
      longueur = str_to_long(strlong);
      if ((longueur % 2)!=0) longueur++;
      if (strncmp (str5,"BMHD",4)==0) Bmhd ();
      else if (strncmp (str5,"BODY",4)==0) Body ();
      else if (strncmp (str5,"CMAP",4)==0) Cmap ();
      else {
         fread(str10K,(int)longueur,1,source);
      }
   }
   fclose (source);

   if (debug) {
      printf ("\ngraph_mode = %d",graph_mode);
      printf ("\nOK");
      getch();
   }

   /* Passage en mode graphique */
   if (Get_graph_mode()!=graph_mode) {
      Set_graph_mode (graph_mode);
      if (Get_graph_mode()!=graph_mode) {
         Set_graph_mode (txt_mode);
         fprintf (stderr,"\nMode graphique %d non autoris\n\n",graph_mode);
         exit (EXIT_ERR_SETGRAPH);
      }
   }

   /* Initialisation de la palette */
   if (use_default_palette==FALSE) {
      switch (graph_mode) {
         case CGA : break;
         case EGA :
         case VGA1 :
         case VGA3 : Init_pal_vga1 (); break;
         case VGA2 : Init_pal_vga2 (); break;
      }
   }


   /* Affichage */
   Datazone_to_display (data_zone);

   /* Fin */
   if (wait_key==FALSE) {
      if (kbhit()) {
         getch ();
         code_sortie = EXIT_KEYPRESSED;     /* option c */
      }
      else {
         code_sortie = EXIT_OK;     /* option c */
      }
   }
   else if (exit_keypressed==TRUE) code_sortie = getch();  /* option 1 */
   else {
      getch (); /* pas d'option */
      code_sortie = EXIT_OK;
   }
   if (keep_graph_mode==FALSE) Set_graph_mode (txt_mode);
   exit (code_sortie);
}
```

```
09/17/1991  09:47        FICHIER  :  OUTIFF.C                    PAGE   4

/*                          BMHD                                  */
/*   ----------------------------------------------------------   */
Bmhd ()
{
   fread(strlong,2,1,source);
   w = str_to_int (strlong);
   fread(strlong,2,1,source);
   h = str_to_int (strlong);
   if (debug) printf ("\nw = %d      h = %d",w,h);
   fread(strlong,2,1,source);
   xo = str_to_int (strlong);
   fread(strlong,2,1,source);
   yo = str_to_int (strlong);
   if (debug) printf ("\nxo = %d      yo = %d",xo,yo);
   fread(&nb_plans,1,1,source);
   if (debug) printf ("\nnb_plans = %d",nb_plans);
   fread(str5,1,1,source);
   fread(strlong,2,1,source);
   compression = str_to_int (strlong);
   if (debug) printf ("\ncompression : ");
   if (debug) {
      if (compression!=0) printf ("OUI");
      else printf ("NON");
   }
   fread(str5,2,1,source);
   fread(str5,2,1,source);
   fread(str5,2,1,source);
   fread(str5,2,1,source);
   fread(str10K,(int)longueur-20,1,source);
   bmhd_lu = TRUE;
   graph_mode = 0;
   txt_mode = 0x03;
   switch (w) {
      case 320 :
         switch (h) {
            case 200 : graph_mode = VGA2; break;
         }
         break;
      case 640 :
         switch (h) {
            case 200 : graph_mode = CGA; break;
            case 350 : graph_mode = EGA; break;
            case 480 : graph_mode = VGA1; break;
         }
         break;
      case 800 :
         switch (h) {
            case 600 : graph_mode = VGA3; break;
         }
         break;
   }
   /* Alloc mmoire */
   data_zone = (unsigned char huge *) halloc (4096+((long)w*(long)h)/2,2);
   if (data_zone==NULL) {
      fprintf (stderr,"\nErreur d'allocation mmoire\n\n");
      exit (EXIT_ERR_ALLOC);
   }
}



/*                          CMAP                                  */
/*   ----------------------------------------------------------   */
Cmap ()
```

```
09/17/1991  09:47        FICHIER  :  OUTIFF.C                          PAGE   5

{
   register int i1;
   register int nb_coul;

   fread(couleur,(int)longueur,1,source);

}



/*                          BODY                                    */
/*   ----------------------------------------------------------    */
Body ()
{
   if (bmhd_lu==TRUE) {
      /* Chargement */
      switch (graph_mode) {
         case CGA : Iff_to_datazone_cga (); break;
         case VGA1:
         case VGA3:
         case EGA : Iff_to_datazone_ega (); break;
         case VGA2 : Iff_to_datazone_vga2 (); break;
      }
   }
   body_lu = TRUE;
}



/*                    IFF_TO_DATAZONE_CGA                           */
/*   ----------------------------------------------------------    */
Iff_to_datazone_cga ()
{
   register unsigned short i1;
   register unsigned short i2;
   unsigned int ptr;
   unsigned int decalage,back;
   char dernier_bloc;
   unsigned char val,mask;
   unsigned char *ptr_buff;
   unsigned int nb_carlu,nbr_bytes;
   int col_en_cours;
   int lgn_en_cours;
   unsigned char huge *ptr_video;

   ptr_video = data_zone;
   dernier_bloc = FALSE;
   while (dernier_bloc == FALSE) {
      ptr = 0;
      if ((nb_carlu=fread(buffer,1,TAILLE_BUFFER,source))!=TAILLE_BUFFER) {
         dernier_bloc=TRUE;
         decalage = 0;
      }
      else decalage = 255;
      while (ptr<nb_carlu-decalage) {
         if (buffer[ptr] > 128) {
            nbr_bytes = 257 - buffer[ptr++];
            val = buffer[ptr++];
            for (i1=0;i1<nbr_bytes;i1++) {
               mask = 0x80;
               for (i2=0;i2<8;i2++) {
                  if ((val & mask)==0) *(ptr_video++) = 0;
                  else *(ptr_video++) = 1;
                  mask >>= 1;
               }
```

```
09/17/1991  09:47        FICHIER  :  OUTIFF.C                          PAGE   6

            }
         }
         else if (buffer[ptr] < 128) {
            nbr_bytes = buffer[ptr++]+1;
            for (i1=0;i1<nbr_bytes;i1++) {
               val = buffer[ptr++];
               mask = 0x80;
               for (i2=0;i2<8;i2++) {
                  if ((val & mask)==0) *(ptr_video++) = 0;
                  else *(ptr_video++) = 1;
                  mask >>= 1;
               }
            }
         }
      }
      back = TAILLE_BUFFER-ptr;
      fseek (source,-(long)back,SEEK_CUR);
   }
}



/*                    IFF_TO_DATAZONE_EGA                           */
/*   ----------------------------------------------------------    */
Iff_to_datazone_ega ()
{
   register unsigned short i1;
   register unsigned short i2;
   unsigned int ptr;
   unsigned int decalage,back;
   char dernier_bloc;
   unsigned char val,mask;
   unsigned char *ptr_buff;
   unsigned char huge *ptr_debut_lgn;
   unsigned char huge *ptr_fin_lgn;
   unsigned char increment;
   unsigned int nb_carlu,nbr_bytes;
   int col_en_cours;
   int lgn_en_cours;
   unsigned char huge *ptr_video;

   if (debug) {
      printf ("\nIN Iff_to_datazone_ega ()");
   }
   ptr_video = data_zone;
   dernier_bloc = FALSE;
   increment = 1;
   ptr_debut_lgn = ptr_video;
   ptr_fin_lgn = ptr_video+w;
   while (dernier_bloc == FALSE) {
      ptr = 0;
      if ((nb_carlu=fread(buffer,1,TAILLE_BUFFER,source))!=TAILLE_BUFFER) {
         dernier_bloc=TRUE;
         decalage = 0;
      }
      else decalage = 255;
      while (ptr<nb_carlu-decalage) {
         if (debug) printf (" %d ",ptr);
         if (buffer[ptr] > 128) {
            nbr_bytes = 257 - buffer[ptr++];
            val = buffer[ptr++];
            for (i1=0;i1<nbr_bytes;i1++) {
               mask = 0x80;
               for (i2=0;i2<8;i2++) {
                  if ((val & mask) == mask) *(ptr_video) += increment;
                  ptr_video++;
```

```
                mask >>= 1;
             }
             if (ptr_video>=ptr_fin_lgn) {
                if (increment<8) {
                   increment <<=1;
                   ptr_video=ptr_debut_lgn;
                }
                else {
                   increment = 1;
                   ptr_debut_lgn = ptr_video;
                   ptr_fin_lgn = ptr_video+w;
                }
             }
          }
       }
       else if (buffer[ptr] < 128) {
          nbr_bytes = buffer[ptr++]+1;
          for (i1=0;i1<nbr_bytes;i1++) {
             val = buffer[ptr++];
             mask = 0x80;
             for (i2=0;i2<8;i2++) {
                if ((val & mask) == mask) *(ptr_video) += increment;
                ptr_video++;
                mask >>= 1;
             }
             if (ptr_video>=ptr_fin_lgn) {
                if (increment<8) {
                   increment <<=1;
                   ptr_video=ptr_debut_lgn;
                }
                else {
                   increment = 1;
                   ptr_debut_lgn = ptr_video;
                   ptr_fin_lgn = ptr_video+w;
                }
             }
          }
       }
    }
    back = TAILLE_BUFFER-ptr;
    fseek (source,-(long)back,SEEK_CUR);
  }
}



/*                    IFF_TO_DATAZONE_VGA2                    */
/*   ------------------------------------------------------   */
Iff_to_datazone_vga2 ()
{
  register unsigned short i1;
  register unsigned short i2;
  unsigned int ptr;
  unsigned int decalage,back;
  char dernier_bloc;
  unsigned char val,mask;
  unsigned char *ptr_buff;
  unsigned int nb_carlu,nbr_bytes;
  int col_en_cours;
  int lgn_en_cours;
  unsigned char huge *ptr_video;

  ptr_video = data_zone;
  dernier_bloc = FALSE;
```



```
  while (dernier_bloc == FALSE) {
    ptr = 0;
    if ((nb_carlu=fread(buffer,1,TAILLE_BUFFER,source))!=TAILLE_BUFFER) {
      dernier_bloc=TRUE;
      decalage = 0;
    }
    else decalage = 255;
    while (ptr<nb_carlu-decalage) {
      if (buffer[ptr] > 128) {
        nbr_bytes = 257 - buffer[ptr++];
        val = buffer[ptr++];
        for (i1=0;i1<nbr_bytes;i1++) {
          *(ptr_video++) = val;
        }
      }
      else if (buffer[ptr] < 128) {
        nbr_bytes = buffer[ptr++]+1;
        for (i1=0;i1<nbr_bytes;i1++) {
          *(ptr_video++) = buffer[ptr++];
        }
      }
    }
    back = TAILLE_BUFFER-ptr;
    fseek (source,-(long)back,SEEK_CUR);
  }
}



/*                      LIRE_ENTETE                           */
/*   ------------------------------------------------------   */
Lire_entete ()
{
  memset (str5,0,5);
  if (fread(str5,4,1,source)!=1) {
    printf ("Err lecture 1 sur %s\n",infile);
    exit (3);
  }
  if (strncmp(str5,"FORM",4)!=0) {
    printf ("Err format 1 sur %s\n",infile);
    exit (4);
  }
  memset (strlong,0,5);
  if (fread(strlong,4,1,source)!=1) {
    printf ("Err lecture 2 sur %s\n",infile);
    exit (5);
  }
  longueur = atol(strlong);
  if (fread(str5,4,1,source)!=1) {
    printf ("Err lecture 1 sur %s\n",infile);
    exit (6);
  }
}



unsigned int str_to_int (unsigned char *str)
{
  return (str[0]*256+str[1]);
}

long str_to_long (unsigned char *str)
{
  long l1;

  return ((long)str[0]*256L*256L*256L+(long)str[1]*256L*256L+
```

```
09/17/1991  09:47        FICHIER  :  OUTIFF.C                    PAGE   9

   (long)str[2]*256L+(long)str[3]);
}



/*                  INIT_PAL_VGA1                          */
/*   ---------------------------------------------         */
Init_pal_vga1 ()
{
   register int i1;

   for (i1=0;i1<16;i1++) {
      Set_Palette_Color (i1,couleur[i1].r,couleur[i1].v,couleur[i1].b);
   }
   return (0);
}


/*                  INIT_PAL_VGA2                          */
/*   ---------------------------------------------         */
Init_pal_vga2 ()
{
   register int i1;

   for (i1=0;i1<256;i1++) {
      Set_Palette_Color (i1,couleur[i1].r,couleur[i1].v,couleur[i1].b);
   }
   return (0);
}




/*          DATAZONE_TO_DISPLAY                          */
/*   ---------------------------------------------         */
Datazone_to_display(ptr_video)
unsigned char huge *ptr_video;
{
   register unsigned short col;
   register unsigned short lgn;

   for (lgn=0;lgn<h;lgn++) {
      for (col=0;col<w;col++) {
         if (use_bios==TRUE) Put_pixel_bios (col,lgn,*(ptr_video++));
         else {
            switch (graph_mode) {
               case CGA : Put_pixel_62 (col,lgn,*(ptr_video++)); break;
               case EGA :
               case VGA1: Put_pixel_63 (col,lgn,*(ptr_video++)); break;
               case VGA2 : Put_pixel_32 (col,lgn,*(ptr_video++)); break;
               case VGA3 : Put_pixel_86 (col,lgn,*(ptr_video++)); break;
            }
         }
      }
   }
}


/*                  PRIMITIVES GRAPHIQUES                        */
/*   ---------------------------------------------------         */
Get_graph_mode()
{
```

```
09/17/1991  09:47        FICHIER  :  OUTIFF.C                    PAGE  10

   union REGS inregs,outregs;

   inregs.h.ah = 0x0F;
   int86 (INT_VIDEO,&inregs,&outregs);
   outregs.h.ah = 0x00;
   return (outregs.x.ax);
}



Set_graph_mode (graph_mode)
unsigned short graph_mode;
{
   union REGS inregs,outregs;

   inregs.h.ah = 0;
   inregs.h.al = (unsigned char)graph_mode;
   int86 (INT_VIDEO,&inregs,&outregs);
   return (0);
}


Put_pixel_bios (x,y,couleur)
unsigned short x,y,couleur;
{
   union REGS inregs,outregs;

   inregs.h.ah = 0x0C;
   inregs.h.al = (unsigned char)couleur;
   inregs.h.bh = 0x00;
   inregs.x.cx = x;
   inregs.x.dx = y;
   int86 (INT_VIDEO,&inregs,&outregs);
   return (0);
}



Set_Palette_Color (num,r,v,b)
unsigned char num,r,v,b;
{
   union REGS inregs,outregs;

   inregs.x.bx = num;
   inregs.h.dh = r>>2;
   inregs.h.ch = v>>2;
   inregs.h.cl = b>>2;

   inregs.h.ah = 0x10;
   inregs.h.al = 0x10;
   int86 (INT_VIDEO,&inregs,&outregs);
   return (0);
}
```

COMPUTER DISCOUNT STORE

LIQUIDATIONS • FIN DE SERIES • SAISIES • MATERIELS DE DEMONSTRATION • SOLDES...

OUVERT 7 JOURS SUR 7 (MEME LE DIMANCHE) DE 11 H A 20 H

LAPTOPS AVEC BATTERIES

- 8088/640 K/2 x 720 K — **3 990 F**
- 80286/640 K/2 x 720 K — **4 990 F**
- 80286/640 K/2 x 720 K/DD 20 Mo — **5 995 F**
- 80286/1 Mo/720 K/DD 40 Mo — **7 995 F**

MONITEURS MONOCHROMES

- 12" Bifréquence — **490 F**
- 14" Bifréquence — **690 F**
- 14" VGA — **850 F**
- Support orientable 14" — **80 F**

IMPRIMANTES NLQ MATRICIELLES

- 130 cps/80 col — **990 F**
- 160 cps/80 col — **1 290 F**
- 480 cps/80 col — **2 690 F**
- 200 cps/132 col — **1 990 F**
- 380 cps/132 col — **2 990 F**

CARTES MÈRES

- 8088 (4,77) — **295 F**
- 80286 (8) — **495 F**
- 80386/SX (16) — **2 490 F**
- 80386 (16) — **2 900 F**

LOGICIELS DE JEUX

- PRINCE — **290 F**
- F15 — **395 F**
- F19 — **395 F**
- RAILROAD TYLOON — **395 F**
- 552 — **345 F**
- GUNS — **345 F**

COMPATIBLE AT 286

1 Mo RAM • Carte + écran bimode • Clavier AZERTY
• Lecteur 1,2 Mo ou 1,44 Mo — **3 390 F**

MONITEURS COULEURS

- 12" VGA — **1 450 F**
- 14" CGA — **1 450 F**
- 14" EGA — **1 750 F**
- 14" VGA — **1 850 F**
- 14" MULTISYNC. — **2 790 F**

LASERS/FAX

- LASER 4 PPM/512 K — **5 995 F**
- LASER 8 PPM/512 K — **8 990 F**
- FAX ITC — **3 495 F**
- FAX CAFAX HS120 — **3 995 F**

GRANDES MARQUES

ONDULEURS

- 250 VA — **1 390 F**
- 350 VA — **1 690 F**

PAO/SCANNERS

- SCAN 200 DPI — **750 F**
- SCAN 400 DPI — **990 F**
- MOUSE — **190 F**
- CLAVIER TRACKBALL — **490 F**

LOGICIELS PROFESSIONNELS

- Intégré ABILITY+ (TT texte, tableur, base de données...) — **450 F**
- DAC EASY LIGHT (comptabilité) — **450 F**
- DAC EASY WORD (traitement de texte) — **450 F**

DISQUES DURS

- 20 Mo/MFM — **990 F**
- 30 Mo/RLL — **1 190 F**
- 40 Mo/IDE — **1 490 F**
- 80 Mo/IDE — **2 595 F**
- FILECARD 30 Mo — **1 595 F**

DIVERS

- Carte ECRAN CGA Hercules — **150 F**
- Clavier AZERTY — **95 F**
- Coffret Rangement 40 x 3"1/2 — **39 F**
- RAM 41256 — **125 F**
- RAM SIPP 256 K x 9 — **90 F**

DISQUETTES (par 100)

- 5"1/4 - 360 K — **1,45 F**
- 5"1/4 - 1,2 Mo — **3,50 F**
- 3"1/2 - 720 K — **3,50 F**
- 3"1/2 - 1,44 MB — **5,90 F**
- 5"1/4 couleurs — **3,25 F**

L'AFFAIRE DU MOIS

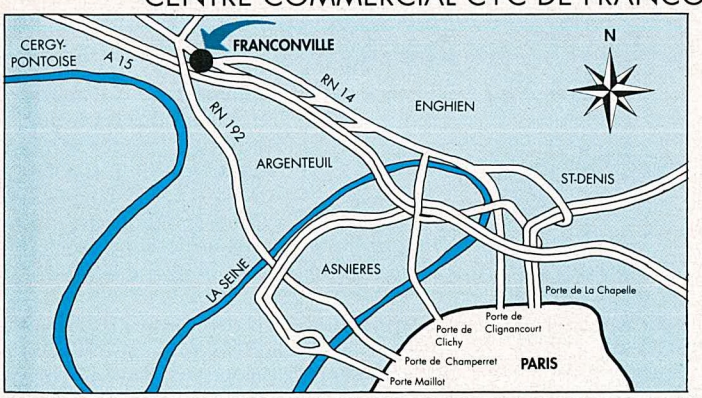
- Carte RAM + 1 Mo installés — **490 F**

Les prix sont donnés Hors Taxe, TVA de 18,6 % non comprise • Offres limitées aux stocks disponibles • Tous nos matériels sont garantis 1 mois, extension possible jusqu'à 1 an (+ 15 %) • Certains matériels peuvent présenter quelques légers défauts d'aspect (rayures, traces, petits accrocs...).

IZARD • RENNES

COMPUTER DISCOUNT STORE

CENTRE COMMERCIAL C+C DE FRANCONVILLE



Rez-de-chaussée

Accès RER : Franconville

(Plessis-Bouchard)

395, av. du Gal. Leclerc

95138 FRANCONVILLE Cedex

Tél. 30.72.55.12

Fax 30.72.55.13

LE COIN DU BRICOLEUR

Matériel présentant des défauts, défaillances, dysfonctionnement, vendu sans garantie - à réviser.

- Streamer 40 Mo Archives* — **490 F**
- Lecteur 5"1/4 / 3"1/2 — **130 F**
- Disque dur 20 Mo — **400 F**
- Moniteur 12" MONO — **250 F**
- Carte RS 232 / parallèle / horloge / jeux à partir de — **60 F**
- Cartes mères / 8088 / 80286 à partir de — **170 F**
- Programmateurs EPROM/PAL, à partir de — **350 F**
- Alimentations 180 W, à partir de — **150 F**

BON DE COMMANDE

A retourner à : COMPUTER DISCOUNT STORE
Centre commercial C+C - Lot 23 - 395, avenue du Général Leclerc - 95138 FRANCONVILLE CEDEX

Nom _____ Prénom _____

Adresse _____

Je commande les matériels suivants :

Description	P.U. H.T.	Qté	Montant net H.T.

Montant total net H.T.

TVA 18,6 %

Montant total net T.T.C.

Ajoutez 70 F de port jusqu'à 5 kg

Ci-joint mon règlement par

☐ chèque ☐ contre-remboursement

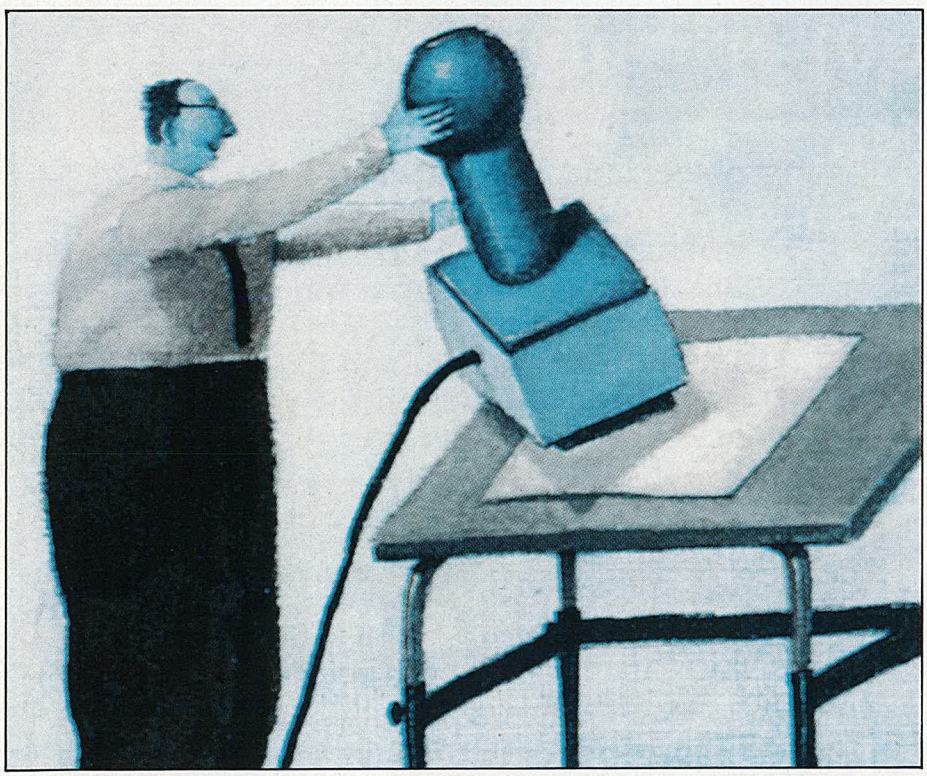
Le prétexte (suite et fin)

Ce programme Prétexte dont la première partie a été publiée dans le n° 71 (Juin-Juillet 1991) de la Revue de l'Utilisateur PC permet de ménager la couche de phosphore de vos tubes cathodiques tout en distrayant votre esprit las et soucieux grâce à de multiples arabesques.

Nous avons vu le mois précédent que notre outil est construit à partir d'un module en assembleur et d'un programme principal en PASCAL. Bien évidemment, les routines de bas niveau se dans la partie en assembleur, qui a été vue dans le numéro 71 de Juin-Juillet 1991 de la Revue de l'Utilisateur PC, et, nous y avons vu comment certaines interruptions BIOS étaient détournées et modifiées, comme par exemple la routine de scrutation permanente du clavier.

En revanche, cette partie en PASCAL que nous publions dans ce numéro contient quant à elle toutes les routines de "haut niveau". C'est dans ce module que nous allons trouver en particulier la routine de pilotage général du programme, ainsi que tous les outils chargés de dessiner nos figures géométriques.

Hugues FERRAND



Le système de pilotage

Nous avons vu dans le numéro de Juin que notre programme détourne 3 interruptions. L'interruption 09 de gestion du clavier, indique au système de pilotage qu'une touche du clavier a été enfoncée.

L'interruption 17h a été enrichie de 2 nouveaux services qui nous permettent d'une part de déterminer si notre programme réside déjà en mémoire, et d'autre part de le désinstaller si nécessaire. Ces deux interruptions détournées et les modules associés se trouvent dans la partie assembleur de notre programme.

La dernière interruption qui est détournée est l'interruption 1Ch. C'est elle

qui constitue le cœur du système de pilotage. Elle est appelée en effet toutes les 18e de secondes. Notre routine teste par l'intermédiaire de la variable globale G_KbmEvent si une touche du clavier a été enfoncée.

Rappelons que c'est l'interruption 09 détournée qui positionne ce flag. Si aucun événement clavier n'a été enregistré, la routine incrémente le compteur d'appels. Si ce compteur atteint une valeur fatidique fixée au lancement de l'application par la ligne de commande, alors la routine sauve le contexte, initialise le mode graphique et positionne le flag G_ArtOn qui indique le traçage des motifs a débuté.

Si en revanche le clavier a été sollicité, alors la routine restitue éventuellement le contexte si le traçage a commencé, et réinitialise le compteur servant au déclenchement du graphique.

Principe du dessin

Le principe de traçage des motifs s'appuie sur un programme fourni en standard avec le package TURBO PASCAL de BORLAND. Il s'agit du programme ARTY.PAS. C'est d'ailleurs ce programme qui nous donné l'idée d'écrire notre outil et le courage de persévérer malgré toutes les embûches rencontrées.

Le principe de base utilisé dans ce programme repose sur ce que l'on peut appeler un rectangle virtuel. Les coordonnées correspondant au coin supérieur et au coin inférieur droit sont gérées dans des variables globales (X1, Y1, X2, Y2).

Ces variables évoluent à chaque fois que la routine de traçage est appelée : leur valeur est modifiée à l'aide d'une variation propre à chaque coordonnée (DeltaX1...). L'affectation des nouvelles valeurs des coordonnées est effectuée dans la procédure AdjustAll.

Une fois les coordonnées du rectangle virtuel calculées, il ne reste plus qu'à tracer à l'intérieur la forme géométrique choisie, à savoir une droite, un cercle, un triangle ou un losange.

Donc, nous savons dessiner un ensemble de figures géométriques dont l'aspect évolue dans le temps.

L'écran en sera cependant rapidement encombré et l'esthétique de l'ensemble deviendra rapidement douteuse. Il faut donc effacer au fur et à mesure les motifs les plus anciens. Il suffit pour cela de stocker dans un tableau (G_Dessin) le type de motif dessiné ainsi que les coordonnées du rectangle virtuel. Lorsque le nombre maximum de dessin admissible sur l'ecran sera atteint, la routine effacera les motifs les plus anciens (procédure DoArt). Et ainsi le tour est joué !

Il est nécessaire de s'attarder un peu sur la procédure chargée du dessin proprement dit des motifs. La première partie de la procédure est construite autour d'un sélecteur CASE. Le dessin du motif est effectué en fonction du type passé en paramètre. Si le code utilisé pour tracer une ligne est évident, celui utilisé pour tracer un cercle l'est moins.

```
Case P_Type OF
    K_LIGNE      :  Graph.Line( X1, y1, x2, y2);
    K_CERCLE     :  Graph.Circle( min( X1, x2) + (ABS( x1-x2) div 2),
                                  min( y1, y2) + (ABS( y1-y2) div 2),
                                  min( 50, max( abs( x1-x2) div 2, ABS( y1-y2) div 2)));
    K_LOSANGE    :  BEGIN
                      DoLosange( X1, Y1, X2, Y2, Losange);
                      Graph.DrawPoly( Sizeof( Losange) DIV Sizeof( PointType), Losange);
                    END;
    K_TRIANGLE   :  BEGIN
                      DoTriangle(X1, Y1, X2, Y2, Triangle);
                      Graph.DrawPoly( Sizeof( Triangle) DIV Sizeof( PointType), Triangle);
                    END;
  END;
```

Ceci mérite une petite explication. Il faut se souvenir en effet que chaque tracé doit tenir dans 1/18e de seconde. Il se trouve que la primitive fournie par BORLAND est assez peu performante, et que la machine sur laquelle le programme a été développé est modeste (compatible 286 à 8 MHz). Si nous n'avions pas pris le soin de brider le traçage du cercle, celui-ci aurait pris trop de temps, ce qui entraîne à (très) court terme l'apparition du message fatal INTERNATIONAL STACK ERROR, SYSTEM HALTED. (Brrrr !).

Le lecteur attentif aura remarqué la syntaxe de l'appel des primitives TURBO PASCAL avec préfixage par le nom de l'unité. Ce type d'écriture évite toute ambiguïté (puisqu'on sait qui est le propriétaire de la routine), et permet de surcharger une primitive déjà existante, tout en se réservant la possibilité d'y faire appel ultérieurement.

Enfin, le programme décrémente à chaque tracé un compteur d'incrément G_IncrementCount. Dès que ce compteur est égal à 0, le programme appelle la routine SelectNewDeltaValues qui modifie les valeurs des déplacements des coordonnées du rectangle virtuel, définit le type de motif à dessiner dans G_TypeDessin, fixe le délai entre deux affichages (G_Delay), et enfin la couleur d'affichage.

Vous remarquerez que le délai d'affichage est extrait d'un tableau indexé par le type du dessin. En effet, une droite est dessinée infiniment plus rapidement qu'un polygone ou un cercle. Il est donc nécessaire d'adapter le ralentissement de la vitesse d'affichage en fonction de sa machine et de ce que l'on veut faire.

Le contexte

Notre programme a la politesse, de travailler proprement. Il doit pour ce faire sauvegarder l'environnement, c'est-à-dire l'écran. C'est la routine InitArt qui est chargée de ce travail. Elle demande d'abord au BIOS quel est le mode graphique en cours, puis sauve cette information ainsi que la mémoire vidéo, la position et la forme du curseur dans la structure de données G_Context.

```
G_Context     : RECORD
                  OldMode  : BYTE;
                  Screen   : ARRAY[0..$3FFF] OF CHAR;
                  Xy       : WORD;
                  CurShape : WORD;
                END;
```

La mémoire vidéo est sauvée dans la variable G_Context.Screen. Nous avons décidé d'effectuer tous nos graphiques suivant la norme CGA, et ce essentiellement pour des raisons d'encombrement mémoire. En effet, la mémoire utilisée par un graphique CGA est de 16 Ko, taille parfaitement compatible avec notre programme résident.
Si nous devions passer en mode EGA ou VGA, il faudrait en revanche mémoriser au minimum 64 Ko. Ce choix reste néanmoins possible, en fonction des options que vous aurez choisies.

Le rétablissement du contexte suit le même principe que la sauvegarde.

La procédure RestoreContext détermine auparavant s'il est nécessaire de réinitialiser le mode vidéo initial. Ainsi, si votre application est déjà en mode graphique CGA, seul le contenu de l'écran sera rétabli. Puis, le curseur est réactivé, et enfin, il est placé sur l'écran là où il était auparavant.

Installation et désinstallation

Le programme démarre par un appel à notre interruption 17h modifiée, afin de déterminer si une copie de notre application n'existe pas déjà en mémoire. Si notre programme a déjà été installé, alors la procédure d'initialisation fait un nouvel appel à l'interruption 17h pour qu'elle lance le rétablissement des interruptions détournées.

Puis, la mémoire occupée par le programme résident est libérée par une mécanique MS-DOS tout à fait classique. Il ne reste plus qu'à informer l'utilisateur que tout s'est bien passé, et qu'il n'oublie pas de bénir le programmeur qui lui a fait un "machin qui tourne sans se planter"

Si notre programme s'aperçoit qu'il est seul sur terre, alors il se lance dans l'analyse de la ligne de commande, pour voir si l'utilisateur lui a laissé quelques instructions. Il faut ensuite qu'il modifie les interruptions que nous utilisons (InitVectors), qu'il rétablisse les vecteurs détournés par TURBO et qu'il informe l'utilisateur que tout est fini.

Quelques remarques techniques

Le driver CGA a été "ojectisé". En effet, il est assez malsain de conserver la mécanique originale de chargement d'un fichier sur disque, d'abord parce que ce fichier peut très bien ne pas être à sa place, et ensuite parce que notre application peut interrompre une interruption MS-DOS, qui comme chacun le sait n'est pas réentrante. D'où un plantage quasiment assuré.

Pour transformer notre CGA.BGI en fichier .OBJ, nous avons lancé l'utilitaire BINOBJ.EXE fourni avec le package TURBO PASCAL.

LISTING

```
C:\>      BINOBJ CGA.BIN CGA.OBJ CgaDriverProc
```


BON DE COMMANDE

à retourner
à la Revue de l'Utilisateur PC
2, rue d'Amsterdam
75009 Paris

Nom : .....
Société : .....
Adresse : .....
Code postal : .....
Ville : .....
Tél. : .....
Montant : 21 F × =
24 F × =
(à partir du n° 45)
Total achat revues =
+ Frais Poste =
Montant du chèque =

Règlement par chèque bancaire :
à l'ordre de SOFTin Communications : 2, rue d'Amsterdam, 75009 Paris
Frais poste pour 1 n° 7,50 F - 2 n° 14,00 F - 3/4 n° 18,00 F - 5/6 n° 25,00 F
(au-dessus par multiple)

POUR COMMANDER LES ANCIENS NUMEROS

Cochez les chiffres qui vous intéressent :

☐ 1	☐ 2	☐ 3	☐ 4	☐ 5
☐ 6	☐ 7	☐ 8	☐ 9	☐ 10
☐ 11	☐ 12	☐ 13	☐ 14	☐ 15
☐ 16	☐ 17	☐ 18	☐ 19	☐ 20
☐ 21	☐ 22	☐ 23	☐ 24	☐ 25
☐ 26	☐ 27	☐ 28	☐ 29	☐ 30
☐ 31	☐ 32	☐ 33	☐ 34	☐ 35
☐ 36	☐ 37	☐ 38	☐ 39	☐ 40
☐ 41	☐ 42	☐ 43	☐ 44	☐ 45
☐ 46	☐ 47	☐ 48	☐ 49	☐ 50
☐ 51	☐ 52	☐ 53	☐ 54	☐ 55
☐ 56	☐ 57	☐ 58	☐ 59	☐ 60
☐ 61	☐ 62	☐ 63	☐ 64	☐ 65
☐ 66	☐ 67	☐ 68	☐ 69	☐ 70
☐ 71	☐ 72			

N° 9 - 17 - 18 - 19 - 20 - 22 - 23 - 34 - 35 - 36 - 37 épuisés

LE SEMINAIRE LDQS
CLIPPER 5.01 DE NANTUCKET

Vous voulez être efficace, performant en 5.01.

Votre emploi du temps ne vous permet pas de vous rendre à la prochaine conférence organisée par Nantucket aux USA.

Les intervenants americains, Craik Yellick, Greg Lief (auteur de GRUMPFISH.LIB), qui animent par ailleurs, toutes les conferences organisées par Nantucket aux USA, vous expliqueront, en français, avec pédagogie 5.01.

1ère SESSION : Débutant - intermédiaire 26 -27 Novembre
2ème SESSION : Avancé 28 -29 Novembre
A Paris, au CNES (Métro les Halles) Agrément formation continue.

Participation par session : 3600 fr h.t. (inscription aprés le 31 Octobre 4100 fr h.t.).

Vous connaissez, de réputation, la qualité de nos séminaires. Ne tardez pas à vous inscrire, nombre de places limité.

* Avant le 31 Octobre 1991

MS DOS M.A.J. GRATUITE AVEC ACHAT DE GRUMPFISH*

L.D.Q.S. vous propose les produits les plus performants du "MONDE CLIPPER" et particulièrement

GRATUIT : vous voulez connaitre la liste commentée des 55 bugs de CLIPPER 5.01, sans obligation d'achat remplissez le bon ci-dessous.

BLINKER 1.5 : en français. Le linker qui a révolutionné la vie des developpeurs. Fini les problèmes de mémoire.

GRUMPFISH. LIB : en français. Livré avec les sources en S'87 et 5.01. Excellent outil pour comprendre 5.01 à travers S'87. Sans modifier votre code incorporer un tableur, agenda, carnet de rendez-vous et plus de 100 fonctions qui vous facilitent la vie (2950 fr h.t. + 50 fr de port).

TELEPATHY. La librairie de Communication (en fran-çais).
Gestion complète des ports séries (fonction pour UART 16550, toutes vitesses jusqu'à 115200 bauds), sans interruption ou non, buffers en émission et réception.
Gestionnaire d'interruptions de timer et série.
Comptabilité : versions S'87 et 5.01, mise en overlays.
Facilité de programmation (2650 fr h.t. + 50 fr de port).

AQUARIUM : Pour 190 fr h.t. par mois: recevez les disquettes ou téléchargez entre 60 et 80 pages de codes, astuces, conseils sur Nantucket 5.01, en français chaque mois. Aucune "newsletter" sur Nantucket ne réunit autant de" Nantucket Gourou".
Du travail de professionnel pour professionnels.
Remboursement au bout de 3 mois si vous n'êtes pas satisfait à 75 %. Pour un abonnement d'un an, nous vous offrons VISIBLE TBROWSE, pour deux ans, un modem 2400 bauds*.

NOTRE SERVEUR : reservé aux abonnés d'aquarium, comprend plus de 400 fichiers d'archives contenant programmes, utilitaires, pour les developpeurs Nantucket.
Toutes les marques citées appartiennent à leur propriétaire respectif.

L.D.Q.S 5, rue Eugene Emanuel
06000 NICE
tél. 93.16.09.12 - fax 93.88.08.90

livraison 24 H

BON DE COMMANDE

NOM .....
SOCIETE .....
ADRESSE.....
CODE POSTAL.....
VILLE.....
TEL.....
FAX.....

- ❑ BUGS 5.01..... GRATUIT
- ❑ GRUMPFISH.LIB..... 2950 FR H.T.+ 50 FR PORT
- ❑ BLINKER 1.05..... 2850 FR H.T. + 50 FR PORT
- ❑ VISIBLE TBROWSE .....1200 FR H.T. + 50 FRPORT
- ❑ TELEPATHY..... 2650 FRH.T. + 50 FR PORT
- ❑ AQUARIUM .....1 AN 2580 FR H.T.
- ❑ .....2 ANS 4560 FR H.T.
- ❑ **SEMINAIRE 5.01.**

Cette ligne de commande aligne le format du driver au format Objet, et déclare la procédure avec le nom public CgaDriverProc. C'est cette déclaration qui est utilisée lors de l'initialisation pour "enregistrer" le driver.

```
if RegisterBGIdriver(@CGADriverProc) < 0 then
    EXIT;
```

Vous remarquerez ensuite que les tailles de la pile et du tas ont été limitées à l'aide de la directive de compilation.

Enfin, tous les contrôles ont été relâchés. Plus de test sur les intervalles ou sur les débordements de pile. Nous travaillons sans filet !

```
{$S-,i-,v-,r-}
{$M 2048,0,5000}
```

Evolutions

Le programme peut évoluer dans le sens d'une modification du moteur graphique. Vous pouvez introduire de nouveaux motifs, modifier le nombre de motifs apparaissant à l'écran (constante Memory), décider de passer en mode EGA ou même VGA, ajouter un menu de paramétrage, dessiner une horloge à la place de motifs géométriques etc...

Il peut aussi servir de canevas pour le développement d'une autre application, par exemple le verrouillage de la machine avec la saisie d'un mot de passe. Vous pouvez même, si vous avez du courage à revendre, vous inspirer de l'utilisation que nous faisons de l'interruption 17h pour écrire un spooler d'imprimante. Et si vous préférez juger sur pièces, il vous suffit de reprendre notre programme et de vous laisser aller à la fascination exercée par la variété des effets graphiques.

LES DISQUETTES DE LA REVUE
GESDOC

Logiciel de recherche documentaire (voir page .....).
Livré sur disquette 3.5" programme et sources utilisable directement ou modifiable par vous-même.
Permet la recherche par auteur, sujet, titre, éditeur, mots-clés,...
Remplace des logiciels qui sont vendus dix fois plus cher...
Une application générée à l'aide de ATELIER !

COUPON
à adresser à La Revue de l'Utilisateur de l'IBM PC
2, rue d'Amsterdam
75009 Paris
Tél. : 47.64.07.57. - 42.67.93.80.

Nom ..... **Prénom** .....
Adresse .....
.....
Code Postal ..... **Ville** .....
Téléphone .....
Veuillez m'envoyer ..... logiciel(s) GESDOC au prix unitaire de 450 F (TTC).
Ci-joint un chèque de ..... F. En règlement à Softin Communications.

Fichier	le 01 Septembre 1991	Page N°
C:\MACLIKE.PAS		1

```
 1: PROGRAM   MACLIKE;
 2:
 3: {$S-,i-,v-,r-}
 4: {$M 2048,0,5000}
 5:
 6: USES CRT, DOS, GRAPH;
 7:
 8: const
 9:    Memory       = 150;
10:    K_WaitFor    :    WORD = 10;  {--- Nb de secondes d'attente ---}
11:    K_LIGNE      =    1;
12:    K_CERCLE     =    2;
13:    K_LOSANGE    =    3;
14:    K_TRIANGLE   =    4;
15:    K_DELAY      : ARRAY[1..4] OF INTEGER = ( 10, 0, 4, 5);
16:
17:
18: TYPE
19:          T_Line     =    RECORD
20:                            DType,
21:                            y1, y2  : BYTE;
22:                            x1, x2  : INTEGER;
23:                          END;
24:
25:          T_CERCLE   =    RECORD
26:                            x1,y1   :    INTEGER;
27:                            Radius  :    INTEGER;
28:                          END;
29:
30:          T_RECTANGLE  = RECORD
31:                            x1,y1   :    INTEGER;
32:                            wd,
33:                            Hg      :    INTEGER;
34:                          END;
35:
36:          T_LOSANGE = ARRAY[1..5] OF PointType;
37:          T_TRIANGLE= ARRAY[1..4] OF PointType;
38:
39:
40: var
41:   ViewXmax,
42:   ViewYmax : integer;
43:
44:   G_Dessin   :  array [1..Memory] of T_LINE;
45:
46:   X1,X2,Y1,Y2,
47:   G_Index,
48:   G_IncrementCount : INTEGER;
49:   DeltaX1,
50:   DeltaY1,
51:   DeltaX2,
52:   DeltaY2       : INTEGER;
53:   GraphDriver,
54:   GraphMode     : integer;
55:   MaxDelta      : BYTE;
56:   G_Delay,
57:   G_Deepth      : INTEGER;
58:   G_TypeDessin : INTEGER;
59:
60:
61: VAR
62:   OldPsp        : WORD;
63:   G_Counter     : WORD;
64:   G_Regs        : REGISTERS;
65:   G_Context     : RECORD
66:                     OldMode : BYTE;
67:                     Screen  : ARRAY[0..$3FFF] OF CHAR;
```

Fichier	le 01 Septembre 1991	Page N°
C:\MACLIKE.PAS		2

```
68:                    Xy      : WORD;
69:                    CurShape: WORD;
70:                  END;
71:   G_ArtOn,
72:   G_KbmEvent    : BOOLEAN;
73:   i             : INTEGER;
74:
75:
76:
77: PROCEDURE CgaDriverProc; external;      {--- driver CGA Objectisé ---}
78: {$L CGA.OBJ }
79:
80: {$L MACLIKE.OBJ}
81:
82: {-------------------------------------------------------------------------}
83: PROCEDURE  AInstall; external;
84: {*************************************************************************}
85: {                                                                         }
86: {    Installe les nouvelles INT09, INT17                                  }
87: {                                                                         }
88: {*************************************************************************}
89:
90: {-------------------------------------------------------------------------}
91: FUNCTION       Min( x, y : integer) : integer;
92: {-------------------------------------------------------------------------}
93: BEGIN
94:   if x < y then
95:      Min := x
96:    else
97:      Min := y;
98: END;
99:
100: {-------------------------------------------------------------------------}
101: FUNCTION       Max( x, y : integer) : integer;
102: {-------------------------------------------------------------------------}
103: BEGIN
104:   if x > y then
105:      Max := x
106:    else
107:      Max := y;
108: END;
109:
110: {-------------------------------------------------------------------------}
111: PROCEDURE        DoLOSANGE( x1, y1, x2, y2 : INTEGER; var P_Losange : T_LOSANGE);
112: {-------------------------------------------------------------------------}
113: BEGIN
114:   P_Losange[1].x := x1;              P_Losange[1].y := (y1+y2) DIV 2;
115:   P_Losange[2].x := (x1+x2) DIV 2;   P_Losange[2].y := y1;
116:   P_Losange[3].x := x2;              P_Losange[3].y := (y1+y2) DIV 2;
117:   P_Losange[4].x := (x1+x2) DIV 2;   P_Losange[4].y := y2;
118:   P_Losange[5]   := P_Losange[1];
119: END;
120:
121: {-------------------------------------------------------------------------}
122: PROCEDURE        DoTriangle( x1, y1, x2, y2 : INTEGER; var P_Triangle : T_TRIANGLE);
123: {-------------------------------------------------------------------------}
124: BEGIN
125:   P_Triangle[1].x := (x1+x2) DIV 2;   P_Triangle[1].y := min( y1, y2);
126:   P_Triangle[2].x := min( x1, x2);    P_Triangle[2].y := max( y1, y2);
127:   P_Triangle[3].x := max( x1, x2);    P_Triangle[3].y := max( y1, y2);
128:   P_Triangle[4] := P_Triangle[1];
129: END;
130:
131:
132: {-------------------------------------------------------------------------}
133: procedure Adjust(var X, DeltaX: integer; ViewMax : INTEGER);
134: {*************************************************************************}
```

Fichier	le 01 Septembre 1991	Page N°
C:\MACLIKE.PAS		3

```
135: {                                                                    }
136: {    Calcule les nouvelles coordonnées des points caractéristiques   }
137: {     d'une droite.                                                  }
138: {                                                                    }
139: { ENTREE  : X       ancienne coordonnée                             }
140: {           DeltaX  variation                                       }
141: {           ViewMax maximum                                         }
142: { SORTIE  : X       nouvelle coordonnée                             }
143: {                                                                    }
144: {*****************************************************************}
145:
146: var
147:   TestX: integer;
148:
149: BEGIN
150:   TestX := X+DeltaX;
151:
152:   IF (TestX<1) or (TestX>ViewMax) THEN
153:     BEGIN
154:       TestX  := X;
155:       DeltaX := -DeltaX;
156:     END;
157:
158:   X := TestX;
159: END;
160:
161: {-----------------------------------------------------------------}
162: procedure AdjustAll;
163: {*****************************************************************}
164: {                                                                 }
165: {                                                                 }
166: {                                                                 }
167: { ENTREE  :                                                       }
168: { SORTIE  :                                                       }
169: {                                                                 }
170: {*****************************************************************}
171: BEGIN
172:    Adjust( x1, DeltaX1, ViewXMax);
173:    Adjust( x2, DeltaX2, ViewXMax);
174:    Adjust( y1, DeltaY1, ViewYMax);
175:    Adjust( y2, DeltaY2, ViewYMax);
176:
177: END;
178:
179: {-----------------------------------------------------------------}
180: procedure SelectNewDeltaValues;
181: {*****************************************************************}
182: {                                                                 }
183: {    Modifie les progressions des points caractéristiques (x,y)   }
184: {     des lignes.                                                 }
185: {                                                                 }
186: { ENTREE  : rien                                                  }
187: { SORTIE  : DeltaX(Y), G_Delay                                    }
188: {                                                                 }
189: {*****************************************************************}
190: VAR
191:   l_Color   : BYTE;
192:
193: BEGIN
194:   DeltaX1 := Random(MaxDelta)-(MaxDelta Div 2);
195:   DeltaX2 := Random(MaxDelta)-(MaxDelta Div 2);
196:   DeltaY1 := Random(MaxDelta)-(MaxDelta Div 2);
197:   DeltaY2 := Random(MaxDelta)-(MaxDelta Div 2);
198:
199:   {--- on force deltaX(y) à une valeur <> 0 ---}
200:
201:   IF DeltaX1 = 0 THEN
```

Fichier	le 01 Septembre 1991	Page N°
C:\MACLIKE.PAS		4

```
202:      DeltaX2 := max( 2, DeltaX2);
203:
204:   IF DeltaY1 = 0 THEN
205:      DeltaY2 := max( 2, DeltaY2);
206:
207:   G_IncrementCount:= Memory;
208:   G_TypeDessin    := max( 1, random( 4));
209:   G_Delay         := random( K_DELAY[ G_TypeDessin]);
210:
211:   {--- Modifie la couleur d'affichage ---}
212:
213:   l_Color := random( 16);
214:
215:   IF l_Color < LIGHTBLUE THEN
216:      l_Color := LIGHTBLUE;
217:
218:   SetBkColor( L_Color);  {-- logiquement SetColor, mais Pb matériel -}
219: END;
220:
221: {-------------------------------------------------------------------------}
222: PROCEDURE InitArt;
223: {*************************************************************************}
224: {                                                                         }
225: {    Initialise le dessin, ainsi que les variables globales utilisées     }
226: {     dans la gestion du dessin.                                          }
227: {                                                                         }
228: {    ENTREE :  rien                                                       }
229: {    SORTIE :  G_Context      Sauvegarde du contexte                      }
230: {              G_Deepth       Nb de ligne du faisceau                     }
231: {                                                                         }
232: {                                                                         }
233: {*************************************************************************}
234: var
235:   i     : integer;
236:   StartX,
237:   StartY: integer;
238:
239: BEGIN
240:
241:   {--- Sauvegarde du contexte actuel ---}
242:
243:
244:   WITH G_Regs DO
245:     BEGIN
246:       Ah := $0F;
247:       INTR( $10, G_Regs);
248:
249:       WITH G_Context DO
250:         BEGIN
251:           OldMode := Al;
252:           Move( MEM[ $B800: 0], Screen, $3FFF);
253:           XY      := WhereX SHL 8 + WhereY;
254:
255:           Ah      := 03;
256:           Intr( $10, G_Regs);
257:
258:           CurShape:= Cx;
259:         END;
260:     END;
261:
262:   {--- Initialisation des aspects graphiques ---}
263:
264:   G_Delay      := random( K_DELAY[ G_TypeDessin]);
265:   G_Index      := 1;
266:   G_TypeDessin := max( 1, random( 4));
267:   G_IncrementCount := 0;
268:
```

Fichier	le 01 Septembre 1991	Page N°
C:\MACLIKE.PAS		5

```
269:   MaxDelta       := 10;  {--- valeur maximale de la variation ---}
270:   GraphDriver    := CGA;
271:   GraphMode      := CGAHi;
272:
273:   InitGraph(GraphDriver, GraphMode, '');
274:
275:   ViewXmax      := GetMaxX;
276:   ViewYmax      := getMaxY;
277:
278:   {--- Initialisation des valeurs des lignes ---}
279:
280:   for I := 1 to Memory do
281:    with G_Dessin[I] do
282:     BEGIN
283:       DType := K_LIGNE;
284:       x1 := 0;
285:       x2 := 0;
286:       y1 := 0;
287:       y2 := 0;
288:     END;
289:
290:    X1 := random( ViewXMax);
291:    X2 := random( ViewXMax);
292:    Y1 := random( ViewYMax);
293:    Y2 := random( ViewYMax);
294:
295:    G_Deepth     := Random( Memory);      {--- fixe le nb de lignes du faisceau ---}
296:
297:    IF G_Deepth < 10 THEN      {--- 10 minimum ---}
298:       G_Deepth := 10;
299: END; {init}
300:
301: {-------------------------------------------------------------------------}
302: procedure UpdatelineCount;
303: {*************************************************************************}
304: {                                                                         }
305: {    Gestion des index utiles au tracé des faisceaux.                     }
306: {                                                                         }
307: { ENTREE  :  G_Index, IncremenCount                                       }
308: { SORTIE  :  G_Index, IncremenCount                                       }
309: {                                                                         }
310: {*************************************************************************}
311:
312: BEGIN
313:   Inc( G_Index);
314:
315:   IF G_Index > G_Deepth THEN
316:      G_Index := 1;
317:
318:   Dec( G_IncrementCount);
319: END;
320:
321: {-------------------------------------------------------------------------}
322: PROCEDURE RestoreContext;
323: {*************************************************************************}
324: {                                                                         }
325: {    Restaure le contexte.                                                }
326: {         Rétablit le mode vidéo si nécessaire                            }
327: {         Rapatrie la mémoire vidéo                                       }
328: {         Repositionne le curseur                                         }
329: {                                                                         }
330: {    ENTREE : rien                                                        }
331: {    SORTIE : rien                                                        }
332: {                                                                         }
333: {*************************************************************************}
334: BEGIN
335:   WITH G_Regs DO
```

Fichier	le 01 Septembre 1991	Page N°
C:\MACLIKE.PAS		6

```
336:     BEGIN
337:       Ah := $0F;
338:       INTR( $10, G_Regs);
339:
340:       {--- Ne restaure le mode vidéo que si nécessaire ---}
341:
342:       IF Al <> G_Context.OldMode THEN
343:          BEGIN
344:            Ah := $00;
345:            Al := G_Context.OldMode;
346:
347:            INTR( $10, G_Regs);
348:          END;
349:
350:       {--- Rétablit la mémoire vidéo ---}
351:
352:       Move( G_Context.Screen, Mem[ $B800:0], $3FFF);
353:
354:       {--- Restaure la forme du curseur ---}
355:
356:       WITH G_Regs do
357:         BEGIN
358:           Ah := 01;
359:           Cx := G_Context.CurShape;
360:
361:           Intr( $10, G_Regs);
362:         END;
363:
364:       {--- Rétablit la position du curseur ---}
365:
366:       GotoXy( G_Context.XY SHR 8, G_Context.XY AND $FF);
367:     END;
368: END;
369:
370:
371: {-------------------------------------------------------------------}
372: PROCEDURE      Draw( X1, y1, x2, y2, P_Color, P_Type : INTEGER);
373: {-------------------------------------------------------------------}
374: VAR
375:   Losange : T_LOSANGE;
376:   Triangle: T_TRIANGLE;
377:
378: BEGIN
379:   SetColor( P_Color);
380:
381:   Case P_Type OF
382:     K_LIGNE      :  Graph.Line( X1, y1, x2, y2);
383:     K_CERCLE     :  Graph.Circle( min( X1, x2) + (ABS( x1-x2) div 2),
384:                                   min( y1, y2) + (ABS( y1-y2) div 2),
385:                                   min( 50, max( abs( x1-x2) div 2, ABS( y1-y2) div 2)));
386:     K_LOSANGE    :  BEGIN
387:                       DoLosange( X1, Y1, X2, Y2, Losange);
388:                       Graph.DrawPoly( Sizeof( Losange) DIV Sizeof( PointType), Losange);
389:                     END;
390:     K_TRIANGLE   :  BEGIN
391:                       DoTriangle(X1, Y1, X2, Y2, Triangle);
392:                       Graph.DrawPoly( Sizeof( Triangle) DIV Sizeof( PointType), Triangle);
393:                     END;
394:   END;
395:
396:
397:   G_Dessin[ G_Index].DType := G_TypeDessin;
398:   G_Dessin[ G_Index].X1    := X1;
399:   G_Dessin[ G_Index].X2    := X2;
400:   G_Dessin[ G_Index].y1    := y1;
401:   G_Dessin[ G_Index].y2    := y2;
402: END;
```

Fichier	le 01 Septembre 1991	Page N°
C:\MACLIKE.PAS		7

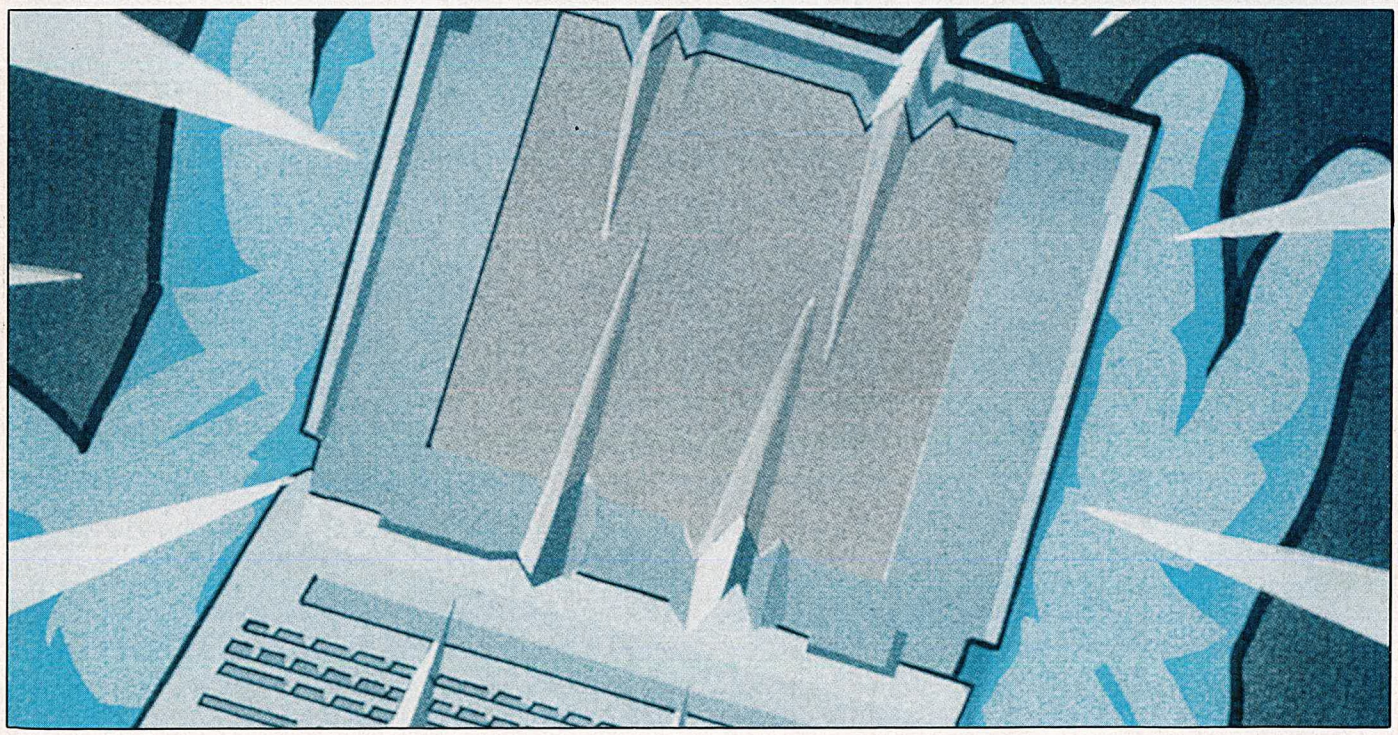
```
403:
404: {-------------------------------------------------------------------------}
405: PROCEDURE DoArt;
406: {*************************************************************************}
407: {                                                                         }
408: {    Trace une droite, et gère le faisceau.                               }
409: {                                                                         }
410: {    ENTREE : rien                                                        }
411: {    SORTIE : rien                                                        }
412: {                                                                         }
413: {*************************************************************************}
414: BEGIN
415:     {--- effacement du motif ---}
416:
417:     WITH G_Dessin[ G_Index] DO
418:       Draw( x1, y1, x2, y2, 0, DType);
419:
420:     IF G_IncrementCount = 0 THEN
421:        SelectNewDeltaValues;
422:
423:     {--- Progression des lignes ---}
424:
425:     AdjustAll;
426:
427:     delay( G_Delay);
428:
429:     {--- Dessin du motif courant ---}
430:
431:     Draw( X1, y1, x2, y2, 1, G_TypeDessin);
432:     UpdatelineCount;
433: END;
434:
435: {-------------------------------------------------------------------------}
436: PROCEDURE NewInt1C;
437: {*************************************************************************}
438: {                                                                         }
439: {    Routine de pilotage du système.                                      }
440: {         Teste si un événement clavier                                   }
441: {         si oui, alors on restaure le contexte                           }
442: {         si non, alors on poursuit le décompte                           }
443: {                                                                         }
444: {    ENTREE : G_Counter       compteur de cycles                         }
445: {             G_KbmEvent      flag événement clavier                     }
446: {             G_ArtOn         dessin en cours                            }
447: {    SORTIE : G_Counter                                                   }
448: {                                                                         }
449: {                                                                         }
450: {                                                                         }
451: {*************************************************************************}
452:
453: INTERRUPT;
454:
455: BEGIN
456:   IF G_KbmEvent THEN
457:      BEGIN
458:        IF G_ArtOn THEN
459:           BEGIN
460:             CloseGraph;
461:             RestoreContext;
462:           END;
463:
464:        G_Counter := 0;
465:        G_ArtOn   := FALSE;
466:        G_KbmEvent:= FALSE;
467:      END;
468:
469:   IF G_ArtOn THEN
```

Fichier	le 01 Septembre 1991	Page N°
C:\MACLIKE.PAS		8

```
470:      BEGIN
471:        DoArt;
472:      END
473:    ELSE
474:      BEGIN
475:        INC( G_Counter);
476:
477:        IF (G_Counter > K_WaitFor*18.2) THEN
478:           BEGIN
479:             InitArt;
480:             G_ArtOn := TRUE;
481:           END;
482:      END;
483:
484: END;
485:
486: {-------------------------------------------------------------------------}
487: PROCEDURE InitVectors;
488: {*************************************************************************}
489: {    Installe les nouveaux vecteurs d'interruption                        }
490: {                                                                         }
491: {         1C  pour le pilotage du système                                 }
492: {         09  pour les événements clavier                                 }
493: {         17  pour désinstaller                                           }
494: {                                                                         }
495: {    ENTREE : rien                                                        }
496: {    SORTIE : rien                                                        }
497: {                                                                         }
498: {*************************************************************************}
499: BEGIN
500:   AInstall;    {--- !!! Obligatoirement en premier ---}
501:
502:   SetIntVec( $1C, @NewInt1C);     {--- installe la nouvelle int 1C ---}
503:
504: END;
505:
506: BEGIN
507:   {--- Test si programme déjà installé ---}
508:
509:   WITH G_Regs DO
510:     BEGIN
511:       {--- appel à INT 17h (détournée) ---}
512:
513:       AX := $FFFF;
514:       Bx := 0;
515:       Intr( $17, G_Regs);
516:
517:       {--- Programme déjà installé -> on le sort de là ---}
518:
519:       IF (Bx <> 0) THEN
520:         BEGIN
521:           OldPsp := Bx;    {--- PSP du programme en mémoire ---}
522:
523:           Ax     := $FFF0;
524:           Intr( $17, G_Regs);    {--- rétablit les anciennes interruptions ---}
525:
526:           {--- Déchargement du bloc d'environnement ---}
527:
528:           G_Regs.AH := $49;
529:           G_Regs.ES := MEM[ OldPsp: $2C]; {--- adresse du bloc d'environnement ---}
530:           MsDos( G_Regs );
531:
532:           {--- Déchargement du programme ---}
533:
534:           G_Regs.AH := $49;
535:           G_Regs.ES := OldPsp; {--- adresse du programme ---}
536:           MsDos( G_Regs );
```

Fichier	le 01 Septembre 1991	Page N°
C:\MACLIKE.PAS		9

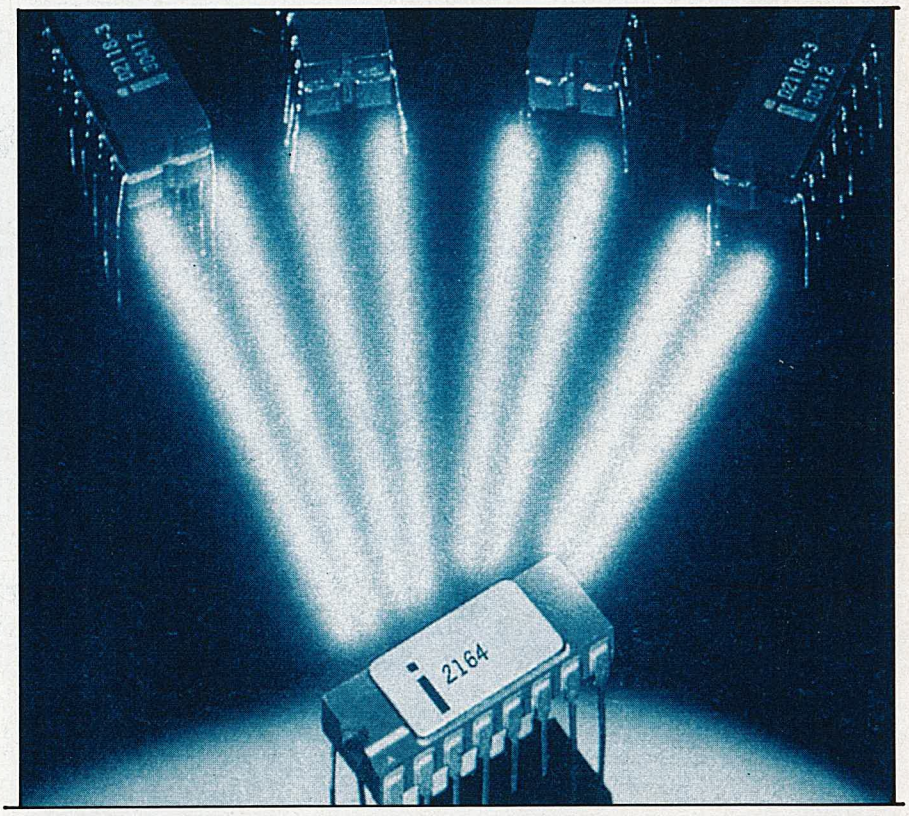
```
537:
538:            TextAttr := $1F;
539:
540:            writeln('                                                ');
541:            writeln('  MACLIKE (c) H. FERRAND                        ');
542:            writeln('    MACLIKE.EXE déchargé de la mémoire          ');
543:            writeln('                                                ');
544:
545:            Halt;     {--- arrêt des hostilités ---}
546:          END;
547:      END;
548:
549:   {--- traitement de la ligne de commande ---}
550:
551:   IF ParamCount > 0 THEN
552:      BEGIN
553:        val( ParamStr(1), K_WaitFor, G_Index);
554:
555:        IF G_Index <> 0 THEN
556:           K_WaitFor := 20;
557:      END;
558:
559:   G_Counter   := 0;
560:   G_ArtOn     := FALSE;
561:   G_KbmEvent  := FALSE;
562:   Directvideo := TRUE;
563:
564:   {--- Teste si le driver est bien là ---}
565:
566:   if RegisterBGIdriver(@CGADriverProc) < 0 then
567:      EXIT;
568:
569:   Randomize;
570:   InitVectors;      {--- détourne les vecteurs d'interruptions ---}
571:
572:   TextAttr := $1F;
573:
574:   writeln('                                                        ');
575:   writeln('  MACLIKE (c) H. FERRAND                                ');
576:   writeln('    Syntaxe : MAKLIKE.EXE [99]     99 = Nb de secondes  ');
577:   writeln('                                          avant réaction');
578:   writeln('    Lancez MACLIKE.EXE pour le décharger de la mémoire  ');
579:   writeln('                                                        ');
580:
581:   SwapVectors:   {--- rétablit les vecteurs détournés par TP ---}
582:   Keep(0);
583: END.
584:
585: {********************** FIN DU SOURCE ***********************************}
```



Programmation Orientée Objet en Assembleur : Une implémentation du Polymorphisme Dynamique

Cet article se propose de montrer comment on peut utiliser les techniques de la Programmation Orientée Objet (POO) en Langage Assembleur ; à la différence des autres langages qui ont été prévus pour cela, ou qui possèdent une surcouche objet, l'Assembleur ne possède pas d'outils spécifiques pour se servir des techniques de la POO.

Paul FRANCESCHI



La démarche consiste donc à fabriquer ses propres outils ; et il existe pour cela des dizaines de méthodes différentes.

Le présent article se propose de réaliser une implémentation du POLYMORPHISME DYNAMIQUE en Assembleur, afin de mettre en évidence comment l'Assembleur autorise cette technique de programmation.

Une des propriétés les plus caractéristiques de ce langage de programmation est en effet l'extrême libreté qu'il laisse au programmeur pour définir ses propres outils, ses propres structures ; tout peut être réalisé en Assembleur, puisque les autres langages ne fournissent en définitive qu'un code en langage-machine, que l'Assembleur peut, dans tous les cas, réaliser.

L'Assembleur peut faire tout ce que peuvent faire les autres langages, mais la réciproque n'est pas vraie. Cependant cette toute puissance est assortie d'une nuance d'importance : il

incombe au programmeur de créer de toutes pièces ses propres outils, ses propres formules. C'est ainsi que le présent article propose une des méthodes d'utilisation du Polymorphisme Dynamique, alors que plusieurs autres demeurent réalisables. Voyons-en maintenant les détails.

Définition

En termes de POO, le Polymorphisme Dynamique s'analyse comme l'aptitude d'une procédure à manipuler des objets de types différents. En programmation de type classique, une procédure fonctionne avec un objet particulier, défini avec ses propres paramètres, toujours identiques ; ceci apparaît comme une limitation, dès lors que l'on souhaite appliquer une même procédure à des objets de types différents.

Le Polymorphisme Dynamique se distingue du Polymorphisme Statique en ce sens que ce dernier effectue la distinction entre les objets de types différents, au moment de la compilation ; à l'inverse, le Polymorphisme Dynamique, beaucoup plus intéressant, réalise la distinction entre les objets en cours d'exécution.

Application

Dans notre exemple d'application, nous nous proposons de réaliser un programme qui utilise trois fonctions fondamentales :

— une fonction d'effacement,
— une fonction d'alignement à droite,
— une fonction d'alignement à gauche.

Ces trois fonctions essentielles et d'usage universel sont nécessaires dès lors qu'on souhaite réaliser un tableur, un traitement de texte, un SGBD, ou un langage de programmation.

Nous désirons également que notre application soit capable d'opérer sur trois types d'objets différents :

— une cellule,
— une ligne,
— une page.

En programmation classique, il serait nécessaire d'écrire les 9 procédures suivantes, correspondant à trois méthodes par objet :

— effacement de cellule,
— effacement de ligne,
— effacement de page,
— alignement à droite de cellule,
— alignement à droite de ligne,
— alignement à droite de page,
— alignement à gauche de cellule,
— alignement à gauche de ligne,
— alignement à gauche de page.

Cette façon de procéder, qui conduit à réaliser un nombre de procédures NP égal à :

NP = Nombre d'objet*Nombre de méthodes

est particulièrement peu économique, et conduit à une inflation démesurée de procédures dès lors que l'on commence à utiliser un nombre de méthodes ou d'objets importants ; or c'est bien le moins que l'on puisse souhaiter dès lors qu'on commence à programmer un projet sérieux.

La POO apporte une solution à ce problème, grâce au Polymorphisme Dynamique, qui permet de réaliser des procédures ou méthodes, qui fonctionnent sur des objets de type différent, et ceci en cours d'exécution. Ceci permet de limiter l'écriture à un nombre de procédures égal à :

NP = Nombre d'objets + nombre de méthodes

On voit que le nombre de procédures à écrire est particulièrement réduit par rapport au schéma classique inflationniste, et que la programmation s'en trouve notablement simplifiée, en même temps que le code nécessaire considérablement réduit.

Pour notre problème particulier, nous devrons écrire trois méthodes multi-objets (MMO) :

— EFFC : effacement,
— ALGN_DRT : alignement à droite,
— ALGN_GCH : alignement à gauche ;

— et il nous sera également nécessaire d'écrire les trois objets utilisés par notre application, qui pourront être appelés par plusieurs procédures ; il s'agit en quelque sorte d'objets multi-méthode (OMM) ;

— CELLULE : l'objet cellule,
— LIGNE : l'objet ligne,
— PAGE : l'objet page.

Implémentation des métodes multi-objets

Nous nous servirons tout d'abord de trois procédures classiques de bibliothèques, qui sont les fonctions primitives, opérant sur une chaîne de caractères pointée par le registre BX, et dont la longueur se trouve placée dans le registre DI. Ces trois routines de base sont :

LISTINGS

```
CODE SEGMENT
;
EFFC:       ; réalise l'effacement d'une chaîne pointée
;           ; par BX et DI
(...)
RET
;
ALGN_DRT:     ; réalise l'alignement à droite d'une
;           ; chaîne pointée par BX et DI
(...)
RET
;
ALGN_GCH:     ; réalise l'alignement à gauche d'une
;           ; chaîne pointée par BX et DI
(...)
RET
;
;
CODE ENDS
;
;
; L'objet METHODE, qui contiendra l'adresse de la méthode
; utilisée.
;
DATA SEGMENT
METHODE    DW    0

DATA ENDS
;
;
```

Lorsqu'un objet appellera une de ces trois procédures, il se chargera de calculer les registres BX et DI, de manière à ce que l'adresse de la chaîne et sa longueur soient calculées de manière correcte. Enfin, nous utiliserons une variable METHODE (adresse de la méthode multi-objets), que le module principal appelant chargera avec l'adresse de l'une de nos trois procédures (EFFC, ALG_DRT, ou ALG_GCH,...).

Implémentation des objets multiméthodes

Nous utiliserons une constante pour chacun des objets, définie avec EQU :

— la constante 1 correspondant à l'objet CELLULE,

— la constante 2 correspondant à l'objet LIGNE,

— la constante 3 correspondant à l'objet PAGE.

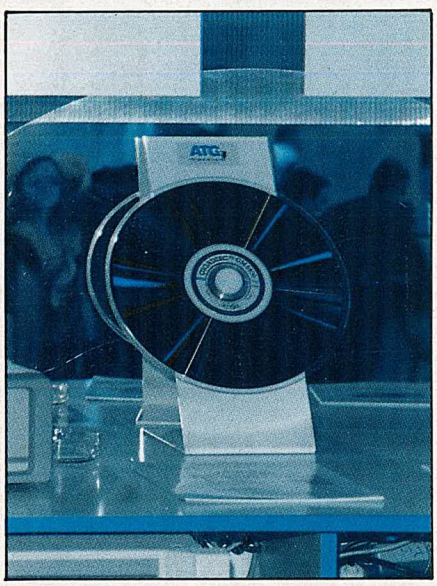
Nous nous servirons ensuite d'une technique originale de définition des objets, qui consiste à les déclarer comme des procédures, et à effectuer ensuite un branchement à leur adresse, lors de l'utilisation de l'objet.

Cette méthodologie est originale, et se démarque de la définition classique d'un objet en POO, qui comporte habituellement :

— des variables (octets, mots, chaînes,...) ;

— des procédures appelées méthodes.

Le fait de définir l'objet comme une constante, qui renvoie à un sous-programme, illustre comment l'infinie liberté du programmeur en Assembleur peut être mise à profit, pour réaliser des outils fonctionnels, en implémentant si nécessaire d'autres modèles que les modèles classiques. En tout état de cause, la méthodologie suivie respecte parfaitement l'esprit et les buts de la POO, puisque le Polymorphisme Dynamique est bel et bien réalisé.



Nous aurons ainsi trois procédures, qui calculent les caractéristiques des objets, c'est-à-dire l'adresse de la chaîne de caractères et sa longueur : 0__CELLULE, 0__LIGNE, et 0__PAGE. Par ailleurs nous nous servirons ultérieurement de la variable OBJET, qui contiendra le code de l'objet utilisé, défini par les constantes ci-dessous :

LISTINGS

```
;
;
CELLULE    EQU   1
LIGNE      EQU   2

PAGE       EQU   3
;
;
DATA SEGMENT
OBJET      DW    0
DATA ENDS
;
;
CODE SEGMENT
;
0_CELLULE:
;
; 1' objet cellule.
; Calcul de BX.
;
(...)
;
; Calcul de DI.
;
(...)
MOV Ax,METHODE
CALL AX
RET
;
;
0_LIGNE:
;
; 1'objet ligne.
; Calcul de BX.
;
(...)
;
; Calcul de DI.
;
MOV AX,METHODE
CALL AX
RET
;
;
0_PAGE:
;
; 1'objet page.
; Calcul de BX.
;
(...)
;
; Calcul de DI.
;
(...)
MOV AX,METHODE
CALL Ax
RET
;
;
CODE ENDS
```

Le noyau du Polymorphisme Dynamique

Voici maintenant la structure du cœur du module d'implémentation du Polymorphisme Dynamique : il s'agit d'une procédure appelée POLY__DYN, qui accepte comme paramètres deux variables, que nous avons déjà évoquées :

— METHODE : l'adresse de la méthode multi-objets,

— OBJET : l'adresse de l'objet multi-méthodes.

En fait, notre module POLY__DIN attend comme paramètres une méthode, et un objet, ensuite de quoi il s'exécute :

Ces quelques routines suffisent à implémenter la totalité du système polymorphique ; il suffit dorénavant de faire varier à volonté les paramètres OBJET (l'objet) et METHODE (la méthode), pour aboutir à l'effet désiré. En cours d'exécution, et à tout moment, votre module pourra changer de méthode, et d'objet.

Un appel à POLY__DIN s'effectuera typiquement de la manière suivante, avec le chargement préalable de METHODE et d'OBJET.

Cet exemple réalise l'effacement d'une ligne.

Ce second exemple effectue l'alignement à gauche d'une page.

Pour optimiser l'écriture des programmes en Assembleur faisant appel fréquemment au noyau du polymorphisme ainsi défini, on pourra se servir de la macro suivante :

Avec cette macro, l'appel au noyau pourra s'effectuer de manière très lisible ci-dessous :

dont l'écriture très limpide se passe de commentaires, mais où l'on constate que l'immense latitude laissée au programmeur en Assembleur pour définir ses propres outils, nous permet d'utiliser une syntaxe plus proche du langage naturel, du type :

POLY ⟨action⟩, ⟨objet⟩

que ne le permettent habituellement les langages orientés objet.

```
CODE SEGMENT
;
POLY DYN:
;
; Le noyau du module de polymorphisme dynamique.
; Les parametres METHODE et OBJET doivent être charges
; avant l appel.
;
MOV  AX,OBJET
; Si l objet cellule est utilise, on appelle le sous-
; programme O_CELLULE .
CMP  AX,CELLULE
JE   O_CELLULE
; si l objet ligne est utilise, on appelle le sous-programme
; O_LIGNE .
CMP  AX,LIGNE
JE   O_LIGNE
; Si l objet page est utilise, on appelle le sous-programme
; O_PAGE .
CMP  AX,PAGE
JE   O_PAGE
; Si aucun objet n est reconnu, on sort du module .
RET
;
;
CODE ENDS
;
```

Les disquettes de la Revue

ARTOOL UN LOGICIEL DE D.A.O. POUR PC ET CARTES EGA OU VGA

ARTOOL possède toutes les fonctions pour exploiter pleinement les possibilités graphiques de votre PC muni d'une carte EGA ou VGA. Il fonctionne en mode BITMAP, et gère toute la palette de couleurs.

Spécifications techniques :

- ☐ Fonctionnement en 16 couleurs, 640×350 pour EGA et 640×480 pour VGA.
- ☐ Dessin à ''main levée'', tracé de droites, de cercles et de boîtes.
- ☐ Ajustement d'une courbe par la méthode de Bézier .
- ☐ Fonctions avancées : loupe, miroir, coloriage, texte.
- ☐ Gestion de la palette, des fichiers images et de l'imprimante.

ARTOOL fonctionne sur tous compatibles PC muni d'une souris et d'une carte graphique EGA ou VGA. Un minimum de 640 Ko ainsi qu'un disque dur sont vivement recommandés. Des impressions graphiques peuvent être obtenues sur des imprimantes compatibles IBM-GRAPH ou EPSON.

Fourni avec les sources en langage PASCAL, vous pourrez adapter ARTOOL à vos besoins et à votre configuration. Une aide technique pourra être apportée par nos soins pour l'exploitation de périphériques non prévus dans la version de base (imprimantes, souris, scanners, etc.).

ARTOOL est disponible sur disquette 5'' 1/4 (360 Ko), au prix de 450 F (TTC)

BON DE COMMANDE
à adresser à La Revue de l'Utilisateur de l'IBM PC
2 rue d'Amsterdam
75009 Paris - Tél. : 42.81.54.27

Nom .....

Adresse .....

Code Postal ..... Ville ..... Téléphone .....

Veuillez m'envoyer ..... logiciel(s) ARTOOL, au prix unitaire de 450 F TTC.

Ci-joint un chèque de ..... F. En règlement à l'ordre de Softin Communications.

```
;
; Cet exemple realise l effacement d une ligne.
;
MOV   AX,OFFSET EFFC
MOV   METHODE,AX
MOV   AX,LIGNE
MOV   OBJET,AX
CALL  POLY_DYN
;
;
;
; Le second exemple efectue l'alignement à gauche d une
; page.
;
MOV   AX,OFFSET ALGN_GCH
MOV   METHODE,AX
MOV   AX,PAGE
MOV   OBJET,AX
CALL  POLY_DYN
;
;
;
```

Pour optimiser l'écriture des programmes en Assembleur faisant appel fréquemment au noyau du polymorphisme ainsi défini, on pourra se servir de la macro suivante:

```
POLY  MACRO     MET,OBJ
MOV   AX,OFFSET MET
MOV   METHODE,AX
MOV   AX,OBJ
MOV   OBJET,AX
CALL  POLY_DYN
ENDM
;
```

Avec cette Macro, l'appel au noyau pourra s effectuer de manière très lisible comme ci-dessous:

```
POLY ALGN_DRT,LIGNE
POLY EFFC,PAGE
POLY ALGN_GCH,CELLULE
POLY EFFC,CELLULE
```

dont l'écriture très limpide se passe de commentaires, mais ou l'on constate que l immense latitude laissée au programmeur en Assembleur pour définir ses propres outils, nous permet d utiliser une syntaxe plus proche du langage naturel, du type:

```
POLY <action>,<objet>
```

que ne le permettent habituellement les langages orientés objet.

Notre exemple ne fait appel, dans un souci de simplification, qu'à 3 méthodes et à 3 objets, mais il est évident que le système peut maintenant croître de manière très simple : il suffit d'ajouter de nouvelles méthodes, et de nouveaux objets.

Les nouveaux objets créés hériteront automatiquement des méthodes des anciens objets : et de même, les nouvelles méthodes ajoutées s'appliqueront d'emblée à tous les objets créés.

Il nous serait désormais aisé d'ajouter les objets suivants :

— colonne,
— curseur,
— reste de la ligne depuis le curseur,
— champ (ensemble de cellules),
— etc...

De même, on pourrait introduire de nouvelles procédures, acceptant comme paramètres BX et DI :

— TOUPPER (conversion en majuscules),

— TOLOWER (conversion en minuscules),

— CENTRE (centrage),

— JUSTI (justification),

— etc...

L'intérêt de l'implémentation du Polymorphisme Dynamique trouve sa pleine justification dès lors que l'on commence à multiplier les objets et les procédures. Il suffit d'écrire le nouvel objet, ou la nouvelle procédure sans tenir compte de ce qui a été écrit, qui est hérité automatiquement.

Le tableau de la figure 1 illustre le nombre de procédures que l'on évite d'écrire en utilisant cette méthode, par rapport à la programmation classique, en fonction du nombre d'objets et du nombre de méthodes utilisés. Ainsi, si l'on utilise comme nous l'avons fait 3 méthodes, et 3 objets, on économise l'écriture de 3 procédures.

Le gain réalisé progresse énormément dès lors que le nombre d'objets et de méthodes s'accroît : c'est ainsi qu'on s'épargne l'écriture de 53 procédures si l'on utilise 10 méthodes et 7 objets. Si par contre, on a besoin de 12 objets et de 13 méthodes, l'économie en procédures atteint le chiffre déjà considérable de 131. Ce qui montre, si besoin était, que les techniques de POO, pour certains projets très structurés, sont désormais indispensables.

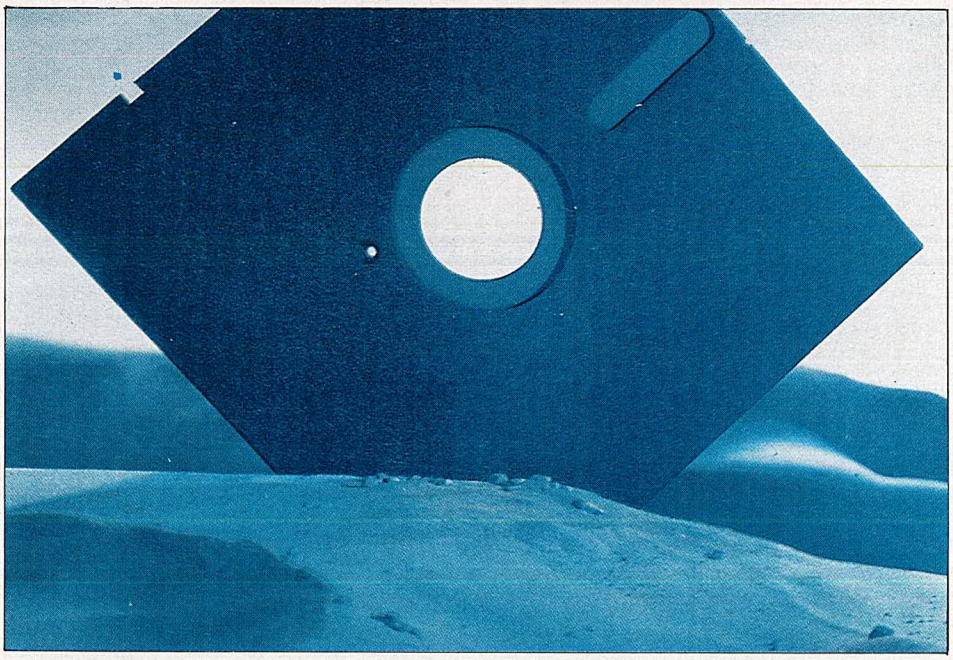
FIGURE 1

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O

	Nombre d' OBJETS										
	1	2	3	4	5	6	7	8	9	10	11	12
Gain en nombre de Procedures												
	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
	-1	0	1	2	3	4	5	6	7	8	9	10
	-1	1	3	5	7	9	11	13	15	17	19	21
	-1	2	5	8	11	14	17	20	23	26	29	32
	-1	3	7	11	15	19	23	27	31	35	39	43
	-1	4	9	14	19	24	29	34	39	44	49	54
	-1	5	11	17	23	29	35	41	47	53	59	65
	-1	6	13	20	27	34	41	48	55	62	69	76
	-1	7	15	23	31	39	47	55	63	71	79	87
	-1	8	17	26	35	44	53	62	71	80	89	98
	-1	9	19	29	39	49	59	69	79	89	99	109
	-1	10	21	32	43	54	65	76	87	98	109	120
	-1	11	23	35	47	59	71	83	95	107	119	131
	-1	12	25	38	51	64	77	90	103	116	129	142

Contrôle d'intégrité d'un disque dur

Patrick POLLET,
Centre Informatique
INSA



Les salles machines de notre Institut sont des lieux privilégiés pour la prolifération des virus.

En effet, en plus de leur utilisation dans la journée pour l'enseignement de l'Informatique, elles sont mises à la disposition des étudiants en libre-service en soirée. Il n'est donc pas rare qu'après une nuit d'utilisation une bonne dizaine de machines soient infestées de Jérusalem, Frodo, ou autre Dark Avenger...

Le problème qui se pose à nous chaque matin est double.

1) Détecter rapidement une éventuelle infection par un virus quelconque.

2) Nous assurer que tous les fichiers nécessaires à l'enseignement n'ont pas été détruits par inadvertance (ou malveillance) pendant la nuit.

Pour le problème des virus, chacun sait qu'il n'y a pas de solutions universelles.

La détection par recherche des signatures (SCAN, UNVIRUS...) est rapide et efficace, mais nécessite une mise à jour périodique pour compléter la collection des signatures recensées.

La méthode par recensement de la taille, de la date et surtout de chaque fichier (NORTON...) est plus satisfaisante, mais elle ne détecterait pas les virus dits "furtifs", (voir plus bas).

Quant à la méthode par contrôles des accès aux fichiers (FLUSHOT...) elle est certes efficace, mais nécessite un temps non négligeable de paramétrage et reste traumatisante pour l'utilisateur qui ne sait pas toujours quoi répondre aux écrans d'avertissements qui apparaissent périodiquement. Et dans ce cas, l'expérience prouve qu'il répond AUTORISER plutôt qu'INTERDIRE.

Enfin, à notre connaissance, aucun détecteur commercial de virus ne permet de vérifier qu'un fichier recensé a disparu du disque.

Nous avons donc écrit un programme maison, qui réalise simultanément les deux tâches : contrôle d'intégrité et test de présence.

Le programme CRCDISK

Ce programme est la réunion de deux modules : l'un pour créer ou mettre à jour un fichier nommé CRCDISK.DTA qui contient une description précise du contenu du disque dur (un enregistrement de type FileCheckRec pour chaque fichier ayant une extension "réputée sensible"), et un second qui reconsulte séquentiellement ce fichier et vérifie que tous les fichiers ainsi recensés n'ont été altérés ni détruits.

Chaque option est lancée par un paramètre sur la ligne de commande : / I pour créer le fichier recensement et / C pour contrôler le disque.

La première option utilise une méthode récursive pour traiter les répertoires et les sous-répertoire inclus (procédure ProcessDir).

Cette dernière appelle répétitivement la procédure ProcessFile qui traite chaque fichier à recenser : calcul et mémorisation de la somme de contrôle, de la date et de la taille du fichier. Notez que nous modifions dans cette procédure la variable FileMode de l'unité System du Turbo Pascal pour que l'appel à Reset() fonctionne aussi sur des fichiers cachés ou en lecture seulement. Sinon le programme planterait immédiatement en tentant d'ouvrir les deux fichiers cachés du DOS (IO.SYS et MsDOS.SYS).

Dans cette version, le programme recense systématiquement les fichiers exécutables réputés sensibles aux virus (EXE, COM et SYS).

Si vous désirez recenser d'autres types de fichiers, il suffit d'ajouter au début de ProcessDir une ligne supplémentaire sur le modèle de celles déjà présentes. Par exemple, pour inclure les fichiers overlays (extension OVR) il suffirait d'ajouter : ProcessFiles (Dir,'OVR').

L'option de vérification relit séquentiellement le fichier CRCDISK.DTA et pour chaque fichier recensé réalise quatre niveaux de contrôle.

— Contrôle d'existence : (code d'erreur 16).

— Contrôle de taille : (code d'erreur 2).

— Contrôle d'heure et de date : (code d'erreur 4).

— Contrôle du contenu : (code d'erreur 8).

En cas d'échec d'au moins un de ces tests, le programme vous avertit par un message et une sonnerie et dépose dans le fichier texte \CRCDISK.RPT un rapport sur le problème. Ce fichier peut ensuite être reconsulté par n'importe quel éditeur de texte.

Le comportement de ce programme peut être modifié grâce à deux directives de compilation conditionnelles définies juste avant la clause USES.

L'option VERIFIE_ABSENCE, qui est active dans notre cas, peut être annulée en insérant un simple point entre l'accolade et le $. Dans ce cas le programme ne signalera plus qu'un fichier recensé a disparu du disque.

En annulant l'option AVEC_RAPPORT, le programme ne produira plus de fichier CRCDISK.RPT et se contentera de messages à l'écran et de sonneries. C'est cette version que nous utilisons normalement. Si un poste sonne, il y a un problème et nous réinstallons immédiatement, sans chercher à comprendre, les logiciels nécessaires à l'enseignement...

Cette réinstallation peut être rendue automatique, puisque le programme se termine en renvoyant à MS-DOS un code d'erreur égal au nombre d'anomalies détectées. Il suffit donc de tester dans un fichier BATCH, la valeur ERRORLEVEL pour déterminer s'il convient de réinstaller. Voici donc l'intérêt de la constante DurId définie dans le programme. Elle permet de tester le contenu du disque dur en étant "sur A:", c'est-à-dire en lançant CRCDISK depuis une disquette de contrôle protégée à l'écriture.

Nous donnons dans le listing 2, un extrait de notre batch de contrôle. La disquettte de contrôle (1.44 Mo) contient le programme CRCDISK, son fichier de recensement et l'ensemble des fichiers nécessaires compactés dans une archive nommée IF2 avec le logiciel du domaine public LHARC. Notez les options /rmc qui permettent :

— /r : reconstituer l'arborescence des répertoires,

— /m : d'inhiber tous les messages d'invite de LHARC (création de répertoire),

— /c : remplacer les fichiers toujours présents... (capital).

Les cas des virus furtifs

Les dernières nouvelles du front semblent indiquer qu'il existerait une nouvelle race de virus, dits furtifs, qui résisteraient à l'observation directe du contenu du fichier infecté. Il n'est pas douteux que CRCDISK serait alors inefficace...

— avoir détourné les interruptions d'accès du disque (21h et surtout 13h) afin de renvoyer à la place du secteur qu'il a infecté, l'ancien contenu qu'il

3615 TEASER

Recevez ***GRATUITEMENT*** *le logiciel BBT pour télécharger avec votre machine* ***(PC XT/AT - ATARI ST - AMIGA - MAC)*** *et venez prendre nos logiciels du domaine public !*

3615 TEASER

Plus de ***10.000 logiciels*** *triés et sélectionnés à votre disposition. Faites votre choix parmi eux. Ils seront chez vous en quelques minutes prêts à l'emploi !*

3615 TEASER

Notre protocole BBT est un des plus rapides (90 cps) et des plus fiables du marché sous Transpac et nos logiciels sont ***les meilleurs et les plus récents.***

3615 TEASER

En quelques minutes chez vous les derniers softs pour ***PC XT/AT, ATARI ST, AMIGA*** *et* ***MAC*** *: tableurs, traitements de textes, langages, graphisme, musique, section adultes et des jeux par milliers.*

Pour recevoir votre **BBT**, adressez à :

FRANCE-TEASER

22, Grande Rue 92310 SEVRES

une disquette vierge avec votre nom, prénom, adresse et type d'ordinateur. Joignez 15 francs en timbres pour frais d'expédition. Vous le recevrez sous 48 h.

Création typon : Club Megaland - 3615 MEGALAND

```
REM exemple de batch de contrôle et ré-installation automatique
ECHO OFF
A:
CRCDISK /C
IF ERRORLEVEL 1 GOTO PROBLEME
GOTO FIN
:PROBLEME
REM les très prudents pourraient appeler ici FORMAT C: !!!!
ECHO patience je ré-installe
LHARC X /rmc A:IF2 C:\
REM si plus d'une disquette de reinstallation
REM compléter ici pour gérer les changements
:FIN
```

listing 2 exemple de batch de contrôle

en avait préalablement sauvegardé. Ceci ne pourrait échapper à un programme d'affichage des résidants comme les logiciels du domaine public SMAP ou MAPSYS... Ce type de logiciel affiche le contenu de la zone système ainsi que les numéros des vecteurs d'interruptions détournés par chaque résidant...

Il suffit donc d'avoir pour chaque poste une copie d'écran des résultats normaux de CHKDSK et SMAP. Un simple contrôle périodique permet de détecter toute manipulation suspecte...

Conclusion

Nous avons présenté ici un petit programme "maison" qui nous permet de contrôler rapidement chaque matin qu'un poste est prêt à son utilisation normale.

Il n'est pas douteux que ce système est certainement perfectible ; mais à ce jour, nous n'avons détecté aucun dysfonctionnement sur les postes qui avaient passé le test avec succès. En ce sens, ce programme remplit parfaitement son rôle qui, rappelons-le, est de rassurer l'utilisateur sans l'encombrer d'avertissements sonores à tout bout de champ...

COUIC

L'outil à désassembler

Que vous **chassiez le virus** ou que vous cherchiez à comprendre comment fonctionne un programme **COM** ou **EXE**, que vous soyez programmeur professionnel ou débutant, il vous faut un bon **désassembleur immédiat**, véritable **fenêtre ouverte** sur le code machine.

— Vous devez pouvoir **désassembler un programme** en ROM, en RAM sur disque ou sur disquette en spécifiant une adresse, **un numéro d'interruption**, un nom de fichier ou un **secteur** et faire défiler le listing de désassemblage ou de dumping en avant et en arrière.

— Vous devez disposer d'instructions **puissantes** de localisation automatique d'**interruption**, de **suivi de jump**, ou de **recherches** de chaînes et de données.

— Vous devez disposer de commandes suffisamment simples pour être apprises et manipulées sans difficulté mais assez efficaces pour faciliter la lecture du code et effectuer des **marquages** dans les zones de données.

— Vous devez pouvoir générer des **fichiers séquentiels** utilisés par les désassembleurs symboliques.

— Vous devez disposer du **code source commenté** de votre désassembleur de façon à l'adapter à vos besoins spécifiques.

COUIC-DESAS permet tout cela et plus encore grâce à ses fonctions spéciales et ses tables descriptives d'interruptions.

Exceptionnellement COUIC est vendu avec ses sources en langage "C" : plus de **9 000 lignes** de code réutilisable dans vos applications et sa documentation pour seulement **340 F (TTC)**.

COUPON
à adresser à La Revue de l'Utilisateur de l'IBM PC
2, rue d'Amsterdam
75009 Paris
Tél. : 47.64.07.57. - 42.67.93.80.

Nom.....

Adresse

.....

Code Postal..... **Ville**

Téléphone.....

Veuillez m'envoyer ..... **logiciel(s) COUIC au prix unitaire de 340 F (TTC).**

Ci-joint un chèque de ..... **F. En règlement à Softin Communications.**

```
program CRCDISK;
{-------------------------------------------------------}
{ AUTEUR : Patrick POLLET INSA}
{ VERSION : 1.0}
{ CREE LE : 02/03/90}
{ DER.MOD.:}
{-------------------------------------------------------}
{ BUT: appel par CRCDISK /i}
{       Construit une table de vérification de touts les fichiers}
{       sensibles à une infection virale (COM,EXE,SYS)}
{    appel par CRCDISK /c}
{       Verifie l'integrité des fichiers sensibles}
{}
{-------------------------------------------------------}

{$DEFINE VERIFIE_ABSENCE}
   {<===signale aussi absence fichiers recensés }
{.$DEFINE AVEC_RAPPORT}
    {<===créer un fichier de rapport si ERREURS  }

uses
 CRT, DOS;

const
 Taille = 25000;
var
 Buffer: array[1..Taille] of byte;

type
 PathStr = string[64];  (* inutile en Turbo >4 *)
type
 ExtStr = string[3];   (* inutile en Turbo >4 *)

type
 FileCheckRec = record      { descripteur d'un fichier sensible}
   Status: LongInt;   { non utilisé }
   Name: PathStr;   { nom complet }
   Time: LongInt;   { date et heure création}
   Size: LongInt;   { taille }
   CRC: LongInt;   { somme de contrôle }
  end;

var
 FileCRC: file of FileCheckRec;    { le fichier de descripteurs }
 ReportFile: Text;

const
 DurId = 'C:';                { le disque a verifie ou que l'on soit }
 FileCRCName = '\CRCDISK.DTA';  { à la racine du disque par défaut }
  {$IFDEF  AVEC_RAPPORT}
 ReportFileName = DurId + '\CRCDISK.RPT'; { à la racine dur disque DUR }
  {$ENDIF}

var
 NbWrongs: Word;
 NbAjouts: Word;
 LastCheckdate:  string;

function CheckSum (FicName: PathStr): LongInt;
{ simple somme de contrôle par ajout des codes hexa }
 var
  F:
  File;
```

```
  CRC: LongInt;
  Nblus, I: word;
begin
 FileMode := 0;        { on prend aussi les fichiers cachés }
 Assign(F, FicName);
 Reset(F, 1);
 CRC := 0;
 while not EOF(F) do
  begin
   BlockRead(F, Buffer, Taille, NbLus);
   for I := 1 to NbLus do
    Inc(CRC, Buffer[I]);
  end;
 Close(F);
 FileMode := 2;      { retour mode normal d'ouverture }
 CheckSum := CRC
end;


(*----------------------------------CREATION DU FICHIER DE CONTROLE ------*)

 procedure ProcessFile (FicName: PathStr; Fic: SearchRec);
{ action sur un fichier }
  var
   Fcr: FileCheckRec;
 begin
  Gotoxy(1, 5);
  Write(FicName);
  ClrEol;
  Inc(NbAjouts);
  FillChar(Fcr, Sizeof(Fcr), 0);
  with Fcr do
   begin
    Status := 0;
    CRC := CheckSum(FicName);
    Size := Fic.Size;
    Time := Fic.Time;
    Name := FicName;
   end;
  Write(FileCRC, Fcr);
 end;


 procedure ProcessFiles (DansDir: PathStr; Ext: ExtStr);
{ traitement de touts les fichiers d'un répertoire }
  var
   Fic: SearchRec;
 begin
  FindFirst(DansDir + '\*.' + Ext, AnyFile, Fic);
  while DosError = 0 do
   with Fic do
    begin
     ProcessFile(DansDir + '\' + Fic.name, Fic);
     FindNext(Fic);
    end;
 end;



 procedure ProcessDir (Dir: PathStr);
{ recursion sur les repertoires }
  var
   SubDir: SearchRec;
```

```
 begin
  ProcessFiles(Dir, 'EXE');
  ProcessFiles(Dir, 'COM');
  ProcessFiles(Dir, 'SYS');
  { vous pouvez ajouter ici d'autres extensions de fichiers sensibles}
  { OVL,OVR,BIN....}
  FindFirst(Dir + '\*.', Directory, SubDir);
  while DosError = 0 do
   with SubDir do
    begin
     if (Name[1] <> '.') and (Attr and $10 = $10) then
      ProcessDir(Dir + '\' + SubDir.name);
     FindNext(SubDir);
    end;
 end;

 procedure CreateCRCFile;
{ option principale /i }
 begin
  Writeln('Création du Fichier de contrôle en cours ... ');
  assign(FileCRC, FileCRCName);
  Rewrite(FileCRC);    { détruit donc l'éventuel ancien }
  NbAjouts := 0;
  ProcessDir(DurId);     {on lance la récursion à la racine du disque DUR}
  Close(FileCRC);
  writeln;
  writeln(' les caractéristiques de ', Nbajouts, ' fichiers sensibles');
  writeln(' aux virus ont été mémorisées ');
  writeln(' Pour verifier l''integrité de votre disque ');
  writeln(' utilisez périodiquement CRCDISK /c ');
 end;

(*-----------------------------------VERIFICATIONS DU DISQUE ------*)

 type
  S15 = string[15];

 function NumStr (N: LongInt; D: Integer): S15;
{ plus facile d'usage que la PROCEDURE Str }
  var
   S: S15;
 begin
  Str(N : D, S);
  NumStr := S
 end;

 function TimeToStr (TD: LongInt): PathStr;
  var
   DT: DateTime;
   H, M, S, Mo, D: string[2];
   Y: string[4];
 begin
  UnpackTime(TD, DT);
  with DT do
   begin
    Str(Hour, H);
    if length(H) = 1 then
     H := '0' + H;
    str(Min, M);
    if length(M) = 1 then
     M := '0' + M;
    str(Sec, S);
    if length(S) = 1 then
```

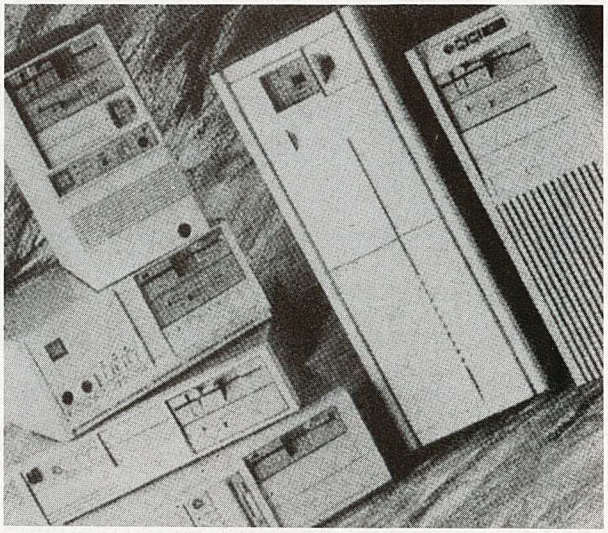
```
    S := '0' + S;
   str(Year, Y);
   str(Month, Mo);
   if length(Mo) = 1 then
    Mo := '0' + Mo;
   str(Day, D);
   if length(D) = 1 then
    D := '0' + D;
  end;
 TimeToStr := +D + '/' + Mo + '/' + Y + ' à ' + H + ':' + M + '.' + S;
end;

procedure Echo (Msg: string);
{ echo ecran et fichier de rapport }
begin
{$IFDEF AVEC_RAPPORT}
 if (NbWrongs = 0) then
  begin
   Assign(ReportFile, ReportFileName);
   Rewrite(ReportFile);
  end;
 Writeln(ReportFile, Msg);
{$ENDIF}
 Writeln ( ^G , Msg );
end;

procedure CheckFile (Fcr: FileCheckRec);
{vérification d'un fichier }
 var
  Sr: SearchRec;
  ErFlag: Byte;
begin
 ErFlag := 0;
 with Fcr do
  begin
   FindFirst(Name, AnyFile, Sr);
   Gotoxy(1, 5);
   Write(Name);
   ClrEol;
   if DosError = 0 then
    begin
     if (Sr.Size <> Size) then
      ErFlag := ErFlag or 2;
     if (Sr.Time <> Time) then
      ErFlag := ErFlag or 4;
     if CRC <> CheckSum(Name) then
      ErFlag := ErFlag or 8;
    end
   else
   {$IFDEF VERIFIE_ABSENCE}
    ErFlag := 16; { ICI ON SIGNALE AUSSI L'ABSENCE...}
   {$ELSE}
   Exit;      { ON PEUT AUSSI NE RIEN FAIRE }
   {$ENDIF}
   if ErFlag <> 0 then
    begin
     GotoXy(1, 10);
     ClrEol;
     echo(Name);
     Inc(NbWrongs);
     if (ErFlag and 2) = 2 then
      echo('    -Taille passée de ' + NumStr(Size, 10) + ' à ' + NumStr(Sr.Size, 10))
     else
```

```
        clreol;
       if (ErFlag and 4) = 4 then
        echo('    -Date passée du ' + TimeToStr(Time) + ' au ' + TimeToStr(Sr.Time))
       else
        clreol;
       if (ErFlag and 8) = 8 then
        echo('    -Le Contenu de ce fichier a été modifié')
       else
        clreol;
       if (ErFlag and 16) = 16 then
        echo('    -Ce fichier a disparu...')
       else
        clreol;
      end;
    end;
end;

procedure Report;
{ rapport final }
begin
 GotoXy(1, 20);
 if NbWrongs = 0 then
  Writeln('aucune anomalie détectée ')
 else
  begin
   Writeln(NbWrongs, ' fichier(s) modifié(s) depuis ', LastCheckdate);
   {$IFDEF AVEC_RAPPORT}
   Writeln(ReportFile, NbWrongs, ' fichier(s) modifié(s) depuis ', LastCheckdate);
   Close(ReportFile);
   Writeln('      Pour obtenir la liste des fichiers modifiés');
   Writeln('      veuillez consulter le fichier \CRCDISK.RPT ');
   {$ENDIF}
  end;
end;

procedure TestCRCFile;
{ arréte le programme si fichier de contrôle introuvable }
 var
  Sr: SearchRec;
begin
 FindFirst(FileCRCName, AnyFile, Sr);
 if DosError <> 0 then
  begin
   Writeln(' Le Fichier de contrôle ', FileCRCName, ' est absent');
   Writeln(' L''avez vous créer une fois avec CRCDISK /i ? ');
   Halt(0);
  end;
 LastCheckDate := TimeToStr(Sr.Time);
 Writeln('Fichier de contrôle crée le ', LastCheckdate);
end;


procedure CheckCRCDisk;
{ option principale /C }
 var
  Fcr: FileCheckRec;
begin
 TestCRCFile;
 Writeln('Vérification en cours ... ');
 assign(FileCRC, FileCRCName);
 Reset(FileCRC);
 NbWrongs := 0;
 while not Eof(FileCRC) do
```

```
    begin
     Read(FileCRC, Fcr);
     CheckFile(Fcr);
    end;
   Close(FileCRC);
   Report;
  end;

  procedure Copyright;
 { un peu de pub...}
  begin
   ClrScr;
   Writeln('PROGRAMME DE VERIFICATION DE L''INTEGRITE D''UN DISQUE');
   Writeln(' (C) Patrick POLLET INSA 1990 ');
  end;

  procedure ShowHelp;
 { si paramétres incorrects}
  begin
   Writeln;
   Writeln('paramétres d''appel incorrects: ');
   Writeln(' CRCDISK /i pour créer une table de vérification du disque');
   Writeln(' CRCDISK /c pour vérifier l''integrité des fichiers EXE,COM et SYS');
   Writeln;
   Halt(0);
  end;


  var
   Pstr: string;

 begin
  NbWrongs := 0;
  CopyRight;
  if paramcount = 0 then
   showhelp;
  Pstr := Paramstr(1);
  if Pstr[1] <> '/' then
   Showhelp;
  case Upcase(Pstr[2]) of
   'I':
     CreateCRCFile;
   'C':
     CheckCRCDisk
   else
     ShowHelp;
  end;
  Halt(NbWrongs);
 end.
```

Que votre volonté soit faite !

Votre micro-ordinateur sur mesure.



Solidement implantée dans le domaine de la fabrication de micro-ordinateurs depuis plus de quatre années, **CB Informatique** vous propose une configuration adaptée à vos besoins demeurant parfaitement évolutive au fil du temps.

Testés durant 72 heures dans nos laboratoires, les micro-ordinateurs CB vous assurent de la qualité professionnelle.

Le coût ? Nous vous en laissons seuls juges !

Voici donc quelques exemples de micro-ordinateurs CB standards pour lesquels nous nous ferons un plaisir de vous proposer un devis correspondant à vos choix.

Tous nos prix s'entendent H.T. départ CB.	Hercules Monochrome	V.G.A. Monochrome	V.G.A. Couleur 800 x 600
CB 286 - 12 Disque dur 40 Mo AT Bus	**5 700**	**6 200**	**7 600**
CB 386 SX - 20 Disque dur 80 Mo AT Bus	**11 200**	**11 650**	**13 900**
CB 386 - 33 Disque dur 110 Mo AT Bus	**16 100**	**16 500**	**19 900**

Toutes ces configurations comprennent : un boîtier 'Desktop', 1 mégaoctet de mémoire centrale, un lecteur 5,25'(1,2 Mo) ou 3,5' (1,44 Mo), les contrôleurs d'écran et de disques, un clavier 102 touches AZERTY ainsi que deux ports série et un port parallèle.

Tarif spécial Revendeurs : nous consulter au tél : (1) 43.96.93.18 ou Fax : (1)43.96.07.44

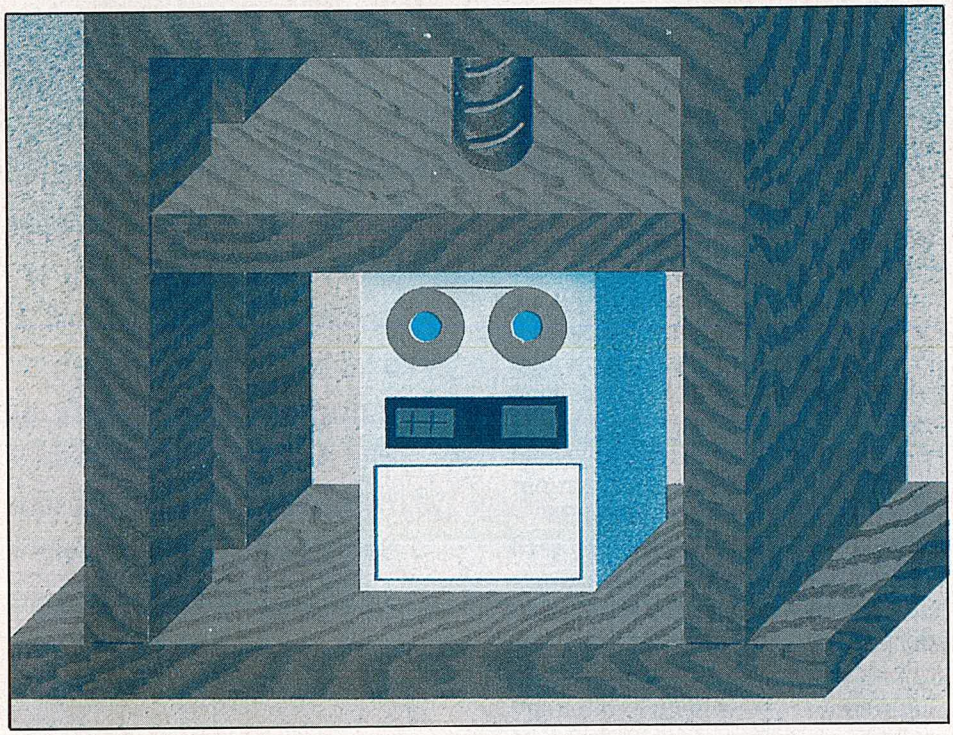
Je souhaite recevoir une proposition concernant le matériel suivant:

Boîtier	Carte mère	Mémoire	Lecteurs	Disque dur	Écran	Souris	MS DOS 4.01	Autres
Desktop Mini-Tower Tower	*286-12, 386 SX-20 386-20, 386-25 386-33*	*1, 2, 4, 8 Mo*	*1 Lecteur 5,25 ou 3,5 2 Lecteurs 5,25 ou 3,5*	*20,40,80,110 Mo ou supérieur*	*Hercules mono VGA mono VGA coul. 800x600*	*Oui Non*	*Oui Non*	*Par exemple : streamer, imprimante, scanner...*

Société : _____
Nom et prénom : _____
Adresse : _____
Code postal : _____
Ville : _____
Téléphone / Fax : _____ / _____

CB informatique
30, avenue du Maréchal de Lattre de Tassigny
94410 SAINT MAURICE
Tél : (1) 43.96.93.18
Fax : (1) 43.96.07.44

WINDOWS 3.0 et la mémoire



Il n'y a pas 1 mais 3 WINDOWS 3.0.

Daniel VASSEUR

Tout le monde parle de WINDOWS 3.0 car tout le monde l'a vu et tout le monde l'utilise en espérant en tirer toute la substantifique mœlle.

Pourtant, bien que ses écrans très reconnaissables s'affichent à l'identique sur tout type de machine. (Bravo VASSEUR, on avait besoin de vous !), la complexité de WINDOWS 3.0 a conduit ses concepteurs à écrire 3 environnements différents.

Pour s'en convaincre, il suffit d'observer et de raisonner un peu.

Mise en évidence

Les utilisateurs actuels de micro-informatique type IBM sont suffisamment avertis pour savoir s'il existe des différences énormes de fonctionnement entre les extrêmes de la gamme de machines standardisées par ce constructeur.

Toujours par observation, et des fois à mes dépens, j'ai découvert des différences de fonctionnement voire des comportements bizarres d'une session sur l'autre et sur une même machine.

Par exemple, j'ai été surpris de constater que je n'arrivais plus à utiliser le presse-papier pour passer une copie d'écran d'une application dans une autre.

De temps en temps je recevais un message d'alerte, ou bien je ne transférais qu'une partie des données sans recevoir aucun message d'erreur.

Dans ce même contexte, j'assistais à une dégradation catastrophique des performances de l'ensemble.

— Le tableur multidimensionnel CA COMPETE ! de COMPUTER ASSOCIATE réclamait des temps de chargement tels que la faculté de médecine m'en a fermement interdit l'utilisation.

— WINWORD — autrement dit WORD pour WINDOWS de MICROSOFT — déroulait alors un menu en un peu plus de 20 secondes !

Après avoir bien recherché, j'ai eu l'idée de jeter un coup d'œil sur mes fichiers CONFIG.SYS et AUTOEXEC.BAT que je savais avoir modifié peu de temps auparavant.

En les consultant, je me suis rappelé que, suite à certains tests, j'avais laissé une configuration système qui n'était pas celle que l'installation automatique de WINDOWS avait mise en place.

Par inactivation de l'ordre de chargement du pilote HIMEM.SYS, j'avais rendu inaccessible toute la mémoire étendue de mon 386SX laissant la machine dans la même configuration qu'un PC/XT.

Si WINDOWS fonctionnait de la même façon quelle que soit la machine sur laquelle il s'exécute, d'une part il serait difficile de trouver une explication aux phénomènes que je viens de décrire, et d'autre part, quel intérêt y aurait-il à posséder une grosse bête 486/33 ou mieux encore ?

Bien que toute ressemblance serait purement fortuite, vous pourriez penser à travers mes propos que je joue à l'idiot du village, aussi vais-je immédiatement anticiper sur la suspicion naissante concernant l'état de facultés mentales : WINDOWS fonctionne selon 3.0 modes différents et pour savoir celui qui est actif, il suffit d'interroger le menu ?. A propos du "Gestionnaire de Programme" du gestionnaire de programme.

On hérite alors d'une petite fenêtre contenant des indications sibyllines mais précieuses sur les possibilités actuelles de l'environnement d'exploitation.

Sans modifier le paramètrage système de ma machine, j'ai forcé le démarrage de WINDOWS de 3 façons distinctes et j'ai copié les vasistas correspondants pour vous les présenter dans les différentes illustrations que je conseille vivement au maquettiste de notre magazine de ne pas omettre d'inclure.

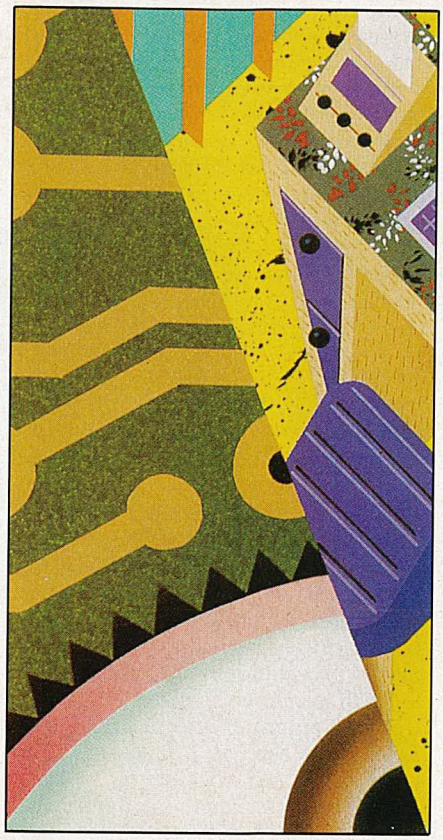
Dans ces reproductions d'écran, et à part les indications relatives au concepteur du produit, ce qui nous intéresse tout d'abord c'est le mode de fonctionnement du logiciel.

La première reproduction fait apparaître la notion de "mode 386 étendu" alors que la seconde boîte d'information parle de "mode standard" et la troisième de "mode réel".

Dans le premier cas, on constate également que la taille de la mémoire disponible est environ 3 à 4 fois plus vaste que la mémoire physiquement installée (le moignon de mémoire conventionnelle signalée en figure n° 3 ne m'intéresse qu'à titre de curiosité personnelle !).

FIGURE 1

FIGURE N°1



Ces constatations étant faites, avançons un peu plus dans les découvertes car ce n'est manifestement pas sans raison que MICROSOFT a donné 3 modes de fonctionnement à son esthéticienne de service.

Les modes de WINDOWS

Allons droit au but : WINDOWS est capable de profiter au mieux des capacités de chacun des microprocesseurs de la famille INTEL, et il dispose d'autant de modes de fonctionnement qu'il y a de générations de microprocesseurs.

Sur un 8086, WINDOWS fonctionne en ''mode réel''. Autrement dit, il sait exploiter la mémoire que DOS lui met à disposition. Je dirai pour simplifier qu'il se comporte comme tout logiciel classique fonctionnant sous DOS.

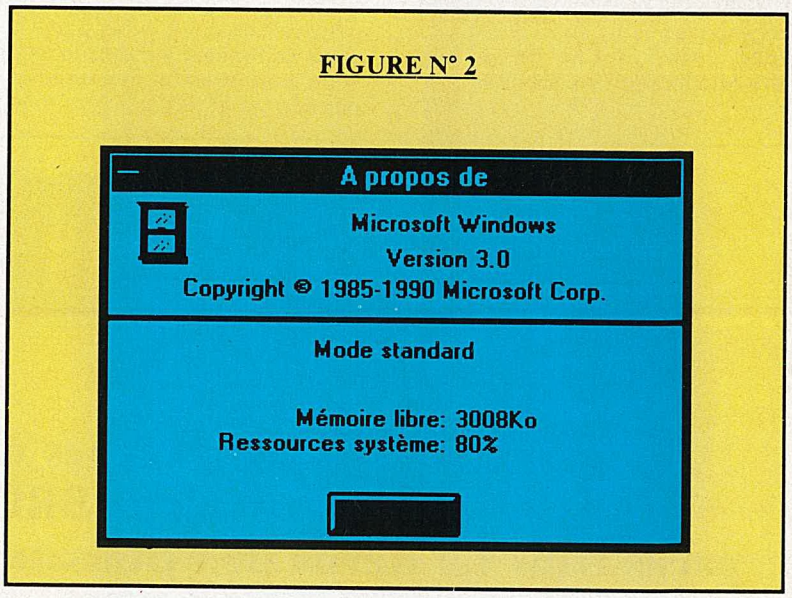
Sur un 80286, par l'intermédiaire du pilote HIMEM.SYS qui est le driver XMS de MICROSOFT, l'accès à la mémoire étendue est possible ce qui dans notre figure n° 2 fait apparaître la disponibilité d'environ 3 MO de mémoire en ''mode standard'' (ma machine est équipée de 4 MO RAM de base).

Sur un 80386, toujours grâce à HIMEM.SYS mais aussi du fait des possibilités de mapping mémoire du microprocesseur, le ''mode 386 étendu'' permet de travailler sur de la mémoire virtuelle, c'est-à-dire de la mémoire provisoirement reléguée sur un disque que l'on souhaite rapide. C'est ainsi que WINDOWS signale la présence de quelque 13 MO de mémoire alors qu'il n'y en a qu'un peu plus de 3 réellement et physiquement disponible.

FIGURE 2

Pour les esprits chagrins qui ne manqueront pas de remarquer que je ne mentionne pas l'existence des 486, je rappelle simplement qu'un 80386 fait exactement la même chose dès lors qu'il est couplé à un 80387 et qu'il est donc hors de question de parler de nouvelle génération.

FIGURE N° 2



Bien entendu, la compatibilité ascendante est assurée, et c'est ainsi qu'un AT/386 (j'insiste !) saura également lancer WINDOWS en mode réel et en mode standard. Seulement quel est l'intérêt du mode réel sur une telle machine ?

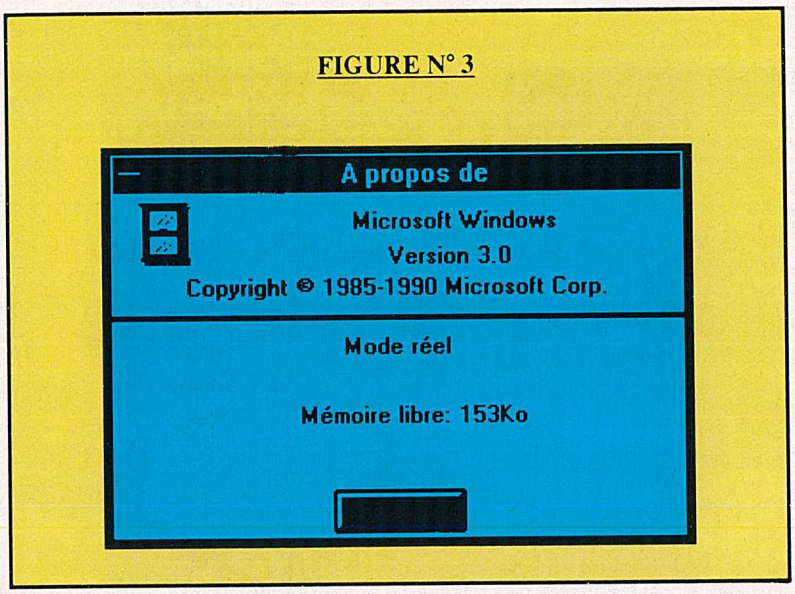
Les avantages des différents modes

Il faut savoir que les possibilités multitâches de WINDOWS existent plus ou moins selon ses modes de fonctionnement. Ce qui fait la différence, ce sont les facilités de paramètrage du noyau temps réel de WINDOWS.

En mode réel, la taille de la mémoire disponible après chargement de WINDOWS — environ 300 KO sous un DOS 5.0 bien optimisé — empêche très vite le lancement de nouvelles tâches. Il existe un remède désormais classique que WINDOWS sait bien entendu utiliser : la mémoire EMS permet d'étendre l'espace de travail de toute machine limitée par les 20 lignes du BUS d'adresses du 8086.

Dans ce cas, la fenêtre d'information confirmera la présence de l'EMS avec, pour les heureux propriétaires de cartes alignées sur la norme EMS 4.0, la notification d'une ''Large Frame'' plus ou moins importante selon l'optimisation de l'installation.

FIGURE 3



Sur un 80286 équipé d'au moins 384 KO de mémoire étendue, et pour autant qu'il trouve un gestionnaire XMS, WINDOWS s'active en mode standard.

Sous ce mode, le fonctionnement d'une tâche en arrière-plan est réservé aux applications écrites pour WINDOWS et des programmes écrits en CLIPPER par exemple n'obtiennent de temps machine que lorsqu'ils sont en avant-plan.

Avec le mode standard, WINDOWS sait accéder à toute la mémoire accessible par le microprocesseur.

Enfin, sur un 80386, le paramètrage du temps microprocesseur à affecter à chaque tâche non WINDOWS se fait dans la mise au point du fichier PIF correspondant, par le biais d'une seconde page de renseignements inaccessibles sous les autres modes de fonctionnement.

Sur ce type de machine, les applications non WINDOWS peuvent également démarrer en mode "fenêtré" par opposition au mode "plein écran" qui est la rançon des applicatifs classiques sous les modes standard et réel, et peuvent donc être "virtualisées" par la désormais célèbre séquence de touches CTRL-ALT-ENTER (se référer à mon précedent article paru dans La Revue de l'Utilisateur PC n° 71, Juin-Juillet 1991).
Comme en mode standard, toute la mémoire gérée par le pilote XMS est prise sous contrôle avec, en plus, mise en œuvre des possibilités propres à ce type de microprocesseur.

WINDOWS et la mémoire physique

Sur une machine 8086, WINDOWS prend sous son contrôle la totalité de la mémoire conventionnelle disponible ainsi que l'éventuelle mémoire EMS mise à disposition par les drivers alignés sur la norme 3.2 et 4.0.
Sur les machines 80286 et pour fonctionner en mode standard, WINDOWS nécessite l'installation d'un pilote XMS au moins aligné sur la version 2.0 de la norme.

En cas de présence simultanée de mémoire EMS et XMS, c'est la seconde solution qui est retenue. Moralité, quand cela s'avère possible, configurez vos cartes d'extension en mémoire étendue !

LES DISQUETTES DE LA REVUE

L'intégralité des programmes des articles Clipper
par Dominique BRUYÈRE
parus dans La Revue de l'Utilisateur PC du n° 66 au 71
avec les codes sources sur disquette
au prix de 280,00 F TTC

COUPON
à adresser à La Revue de l'Utilisateur PC
2, rue d'Amsterdam
75009 Paris
Tél. : 47.64.07.57. - 42.67.93.80.

Nom .....
Adresse .....
.....
Code Postal ..... **Ville** .....
Téléphone .....
Veuillez m'envoyer ..... logiciel(s) Clipper au prix unitaire de 280,00 F TTC.
Ci-joint un chèque de ..... F. En règlement à Softin Communications.

Avec le mode standard, les limites sont repoussées très loin. Sous ce mode, WINDOWS peut gérer, en plus de la mémoire conventionnelle, les 15 MO de mémoire étendue que peuvent supporter presque toutes les machines de type AT/286. Mais attention, ce n'est pas forcément une solution idéale !

Rares sont les machines de type 286 qui acceptent une large plage de mémoire étendue sur carte-mère et qui obligent donc à faire appel à des cartes d'extension qui s'installent sur l'un des slots du BUS de la machine.

Outre l'obstacle du coût, il faut savoir que la mémoire d'extension sur carte additionnelle est beaucoup moins rapide (environ trois fois sur un 286 à 16 MHz) que celle directement implantée sur carte-mère, et que les performances de WINDOWS s'en trouvent naturellement affectées.

Mais, plus grave, les pilotes XMS gèrent très mal des disparités aussi grandes de temps d'accès ce qui se traduit le plus souvent par le plantage pur et simple de la machine.

C'est d'ailleurs pour régler en partie ce problème que j'ai rédigé les deux utilitaires XMSBLOQ et XMSFREE parus dans ces mêmes colonnes en Septembre 1991, n° 72 de La Revue de l'Utilisateur PC.

Ce problème des temps d'accès à la mémoire étendue des 286 se trouve décalqué à l'identique sur les machines de type 386SX qui n'acceptent le plus souvent que 4 MO de RAM sur carte-mère et le reste sur carte d'extension. Pour ces machines, les différences fonctionnelles entre modes standard et étendu se situent donc uniquement sur le terrain de la mémoire virtuelle et celui du fonctionnement multitâches des applications non WINDOWS.

Sur les machines 80386, la mémoire EMS physique est totalement inutile. La présence d'un pilote XMS sera, là aussi, seule suffisante pour provoquer le lancement du mode 386 étendu.

De toutes façons, WINDOWS en mode 386 étendu simule la présence de mémoire EMS, sauf sur ma propre machine où, bien que capable d'en détecter toutes les caractéristiques, je n'ai jamais réussi à en obtenir une seule page malgré de très nombreux essais.

A l'évidence, WINDOWS trouve sa meilleure plate-forme d'installation sur ce type de machine largement pourvue en mémoire (4 MO est un strict minimum, 8 ou 16 MO ne doivent pas faire peur !).

Pour éviter tout problème, il convient au moment du choix matériel, de retenir des machines en fonction de leur aptitude à engranger une grande quantité de mémoire sur carte-mère - de nombreux 486 y comptent couramment 64 MO - et ne pas hésiter à intégrer un coprocesseur arithmétique pour des raisons de performances (INTEL vient de baisser ses prix de façon remarquable, – 70 %, pour lutter contre la concurrence grandissante).

Enfin, pour répondre à la question "Combien faut-il installer de mémoire ?", je répondrai diplomatiquement qu'il vaut mieux en mettre suffisamment pour éviter le swapping sur disque, cause d'une dégradation très sensible des performances de l'ensemble.

Les programmes et la mémoire de WINDOWS

Vous savez désormais comment WINDOWS s'alimente avec la mémoire de votre machine. Regardons maintenant comme il la restitue aux programmes qu'il a pour mission de fédérer.

Il convient de distinguer les programmes classiques des applications WINDOWS.

Les programmes classiques se voient attribuer autant de mémoire conventionnelle que WINDOWS peut leur en dégager. Autant dire tout de suite que dans les configurations complexes comme les réseaux locaux, la portion disponible est dramatiquement réduite du fait de toutes les couches logicielles résidantes chargées préalablement au lancement de WINDOWS.

En plus de tout cela, par l'intermédiaire des fichiers PIF il est possible d'affecter de la mémoire XMS et/ou EMS à toutes les applications qui savent en tirer profit. Un programme CLIPPER 5.01 linké avec BLINKER 1.5 tournera donc sans difficultés dans une tâche WINDOWS avec des performances convenables.

Se pose par contre le problème des applications qui utilisent la mémoire étendue de façon sauvage : FRAMEWORK III en fait partie et la seule solution pour garantir la cohabitation passe par le surcoût de l'achat et installation d'un gestionnaire de mémoire.

Pour les applications WINDOWS, le problème est différent et l'on serait tenté de croire qu'elles devraient toujours fonctionner quel que soit le mode d'exécution de l'environnement hôte.

Il arrive pourtant que, du fait de la complexité et des difficultés de mise au point des programmes, des éditeurs de logiciels court-circuitent délibérément certains modes de fonctionnement.

C'est ainsi que les TURBO PASCAL et DEBUGGER POUR WINDOWS de BORLAND ne savent pas travailler sous un WINDOWS en mode réel et refusent tout simplement de se lancer.

D'autres programmes ne disent rien mais ne font rien. Tel est le cas d'EVOLUTION le traitement de texte d'EVOLUTION qui ne sait pas faire évoluer son correcteur orthographique en mode réel.

Comme on s'en aperçoit — souvent trop tard —, le label "Compatible WINDOWS 3.0" n'est pas forcément un gage de satisfaction, à un point tel que dans mon précédent article, j'avais parlé de "miroir aux alouettes du mode réel"

Dans d'autres cas tout aussi embarrassants, c'est l'insuffisance de mémoire qui empêche le lancement des programmes pourtant bien écrits. C'est ainsi que des poids lourds du traitement de textes comme WINWORD de MICROSOFT refusent obstinément de démarrer sur des configurations trop étroites.

Le mode réel strict leur est pratiquement interdit du fait de leurs exigences en ressources mémoire.

De toutes façons, s'ils arrivent à démarrer, c'est l'utilisateur qui craque nerveusement devant leurs nouvelles performances !

Au chapitre de la programmation, il faut savoir que WINDOWS est seul responsable de la gestion de la mémoire globale disponible pour toutes les applications, et qu'à ce titre il lui arrive de réorganiser l'implantation des plages mémoires distribuées.

Cette opération, connue sous le terme de "compactage", oblige les programmes en cours d'exécution à ne jamais présumer de l'adresse de départ d'une zone allouée.

Il ne faut pourtant pas s'imaginer que programmer sous WINDOWS pose des problèmes insurmontables : il suffit simplement de se plier à quelques règles d'hygiène élémentaire que je détaillerai une autre fois.

Quel que soit le mode d'exécution de WINDOWS, un programme peut utiliser 2 réservoirs distincts de mémoire : le tas local et le tas global.

Selon le compilateur et le modèle mémoire choisi, les limitations imposées à la taille du tas local — 64 KO avec TURBO PASCAL POUR WINDOWS de BORLAND — peuvent être pénalisantes pour le programme qui doit alors faire appel au tas global dans lequel des zones allant jusqu'à 1 MO en mode standard et 64 MO en mode 386 étendu peuvent être prélevées.

Voici, si besoin s'en faisait encore sentir, une nouvelle confirmation que WINDOWS donne aux programmes des possibilités de travail auparavant difficilement accessibles.

Conclusion

L'interface utilisateur de WINDOWS séduit énormément de par ses possibilités graphiques et sa facilité d'emploi.

Un peu comme la partie émergée de l'iceberg, les jolis écrans graphiques ne doivent pas nous faire oublier la complexité de mise au point qu'elle a déjà imposée aux services de développement de MICROSOFT qui se sont vus contraints d'écrire 3 noyaux différents pour profiter au mieux des capacités respectives des 3 générations de matériels actuellement représentatives du marché de la micro.

A la lueur de ma récente expérience de programmeur WINDOWS, je peux prédire sans me tromper que cet environnement demandera une somme de travail plus grande encore à l'armée des programmeurs désireux d'en profiter.

Heureusement pour nous, les grands éditeurs de logiciels ont déjà peaufiné des panoplies du parfait petit développeur sous WINDOWS, et il y en a pour tous les goûts.

Je me contenterai de citer :

— OBJECT WINDOWS de BORLAND fournit en français depuis Juillet 1991 dans le package du TURBO PASCAL POUR WINDOWS et qui devrait être également disponible pour TURBO C au moment où vous lirez ces lignes ;

— VISUAL BASIC de MICROSOFT qui est d'une simplicité d'emploi que je qualifierai de biblique.

Indépendamment de votre choix parmi ces 2 produits et les autres disponibles sur le marché, il vous faudra faire l'effort de parler ''objet'' ce qui semble être à l'heure d'aujourd'hui la seule façon de ne plus se perdre dans le bourbier de la maintenance des développements mal conçus à l'origine.

Après le raz-de-marée des langages structurés, les développeurs chenus sont donc confrontés à une seconde mutation d'importance : le raisonnement orienté objet.

De plus, WINDOWS introduit en force la notion de programmation événementielle : mais que nous réserve donc encore l'avenir de cette informatique galopante ?

INTERGREZ LA PUISSANCE DES GRAHIQUES DANS VOS APPLICATIONS.

PCX PROGRAMMER'S TOOLKIT 4.0

Un outil de développement qui vous permet d'inclure rapidement et facilement des graphiques dans vos applications. Le PCX Programmer's Toolkit comprend plus de 90 routines permettant l'affichage, la sauvegarde,le redimensionnement et l'impression des graphiques PCX.

2490 TTC

GX EFFECTS 2.0 *Nouveau!*

Animez vos programmes en y incluant des effets spectaculaires: effacement, "effet de brisure", compression, glissement d'image, image en diagonale et explosion d'image...Ajouter à cela, la possibilté de rajouter des effets musicaux grâce au langage musical de GX Effets.

1990 TTC

GX TEXT 2.0 *Nouveau!*

Avec GX Text vous aurez la possibilité d'inclure du texte à vos graphique. De nombreux types de caractères sont inclus, mais cependant vous aurez toujours la possibilité de créer d'autres types de caractères avec l'éditeur de fontes.

1490 TTC

GX GRAPHICS 1.0

Un ensemble complet de routines comprenant toutes les primitives graphiques (sont incluses les primitives graphiques des libraries graphiques Borland (BGI)et Microsoft) . Il supporte de nombreux modes (Hercules, CGA, EGA, VGA, Super VGA). GX Graphics comprend plus de 100 routines écrites en Assembleur.

1990 TTC

Langages supportés: MS Assembleur 5.x, 6.0, Turbo Assembleur 1.0, 2.0, MS C 5.x, 6.0, Quick C 1.x, 2.x, Turbo C 1.x, 2.0, Lattice C, JPI Topspeed C 3.0, Borland C++2.0, JPI Topspeed C++ 3.0, MS Pascal 4.0, Turbo Pascal 4.0, 5.x, 6.0, JPI Topspeed Pascal 3.0, MS Basic 6.0, 7.x, MS Fortran 4.1et 5.0.

Les produits

G·E·N·U·S

MICROPROGRAMMING

sont distribués officiellement en France par

VERSION US

17, Avenue Emile Zola, 75015 Paris

Tél: 33 (1) 40 59 09 13

Si vous êtes déjà un utilisateur de CLIPPER de Nantucket, vous avez besoin du

Générateur de programme
ATELIER

Il fait pour vous :

☐ la génération du menu principal,
☐ la génération de sous-menus,
☐ la génération des programmes de saisie,
☐ la génération des programmes d'impression,
☐ la gestion de vos sources,
☐ le lancement du compilateur...

ATELIER

☐ incorpore votre copyright et une calculette dans vos programmes
☐ génère le code fastidieux (90 %)
☐ et vous laisse faire la partie intéressante de la programmation.

ATELIER

☐ est livré avec les sources commentées, ainsi qu'une documentation imprimable.

La version compilée n'est pas livrée sur la disquette car ATELIER est destiné aux utilisateurs de CLIPPER et nécessite ce compilateur.

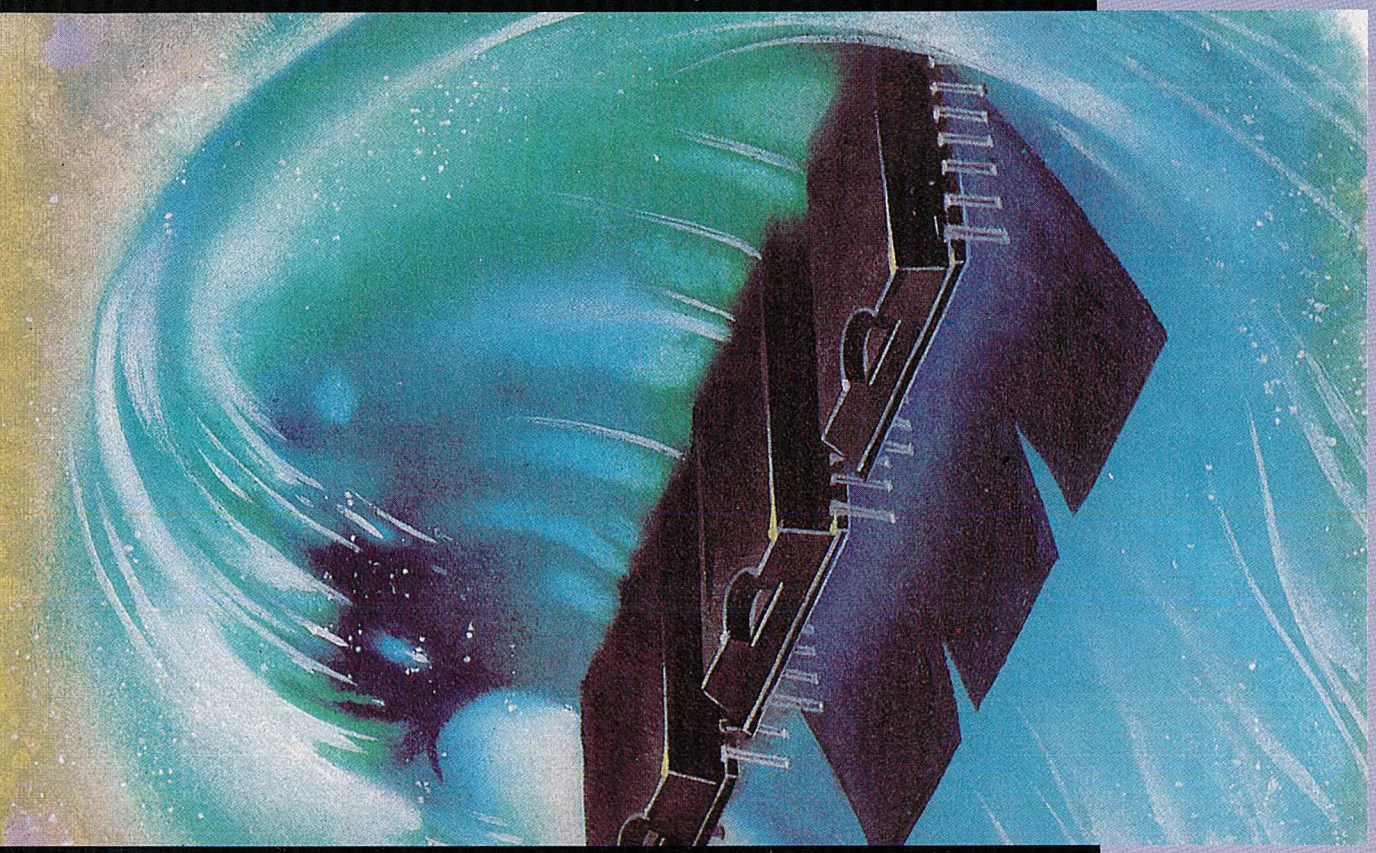
Les commentaires sur ce programme ont été publiés dans La Revue de l'Utilisateur de l'IBM PC (numéro 59, Janvier-Février 1990), et la suite de la série comportera 6 articles à compter du numéro 61, Avril 1990.

ATELIER

est disponible auprès de La Revue de l'Utilsateur de l'IBM PC au prix de 450 F, sur disquette 5''1/4 ou 3''1/2.

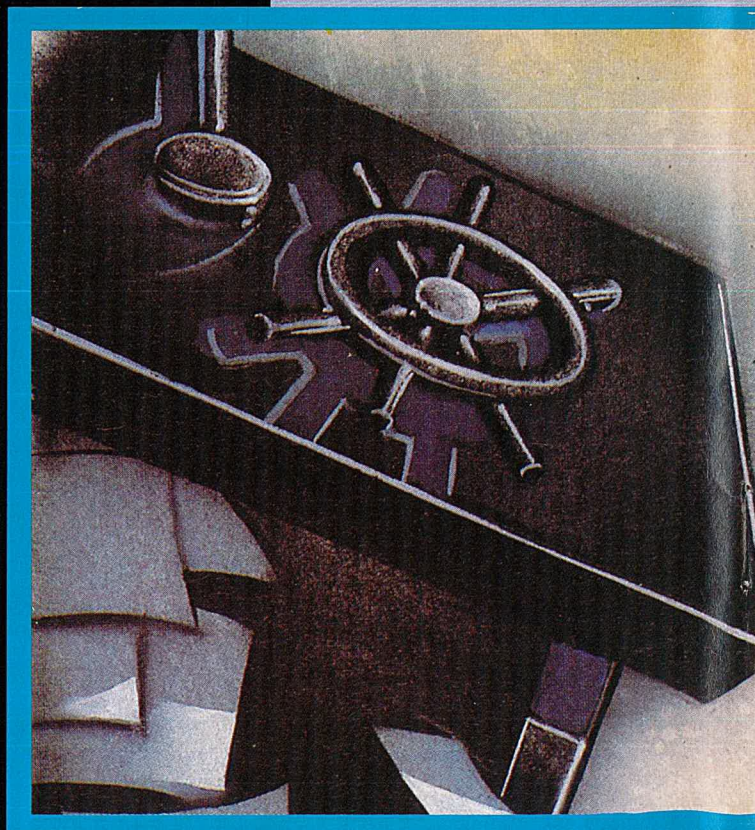
COUPON
à adresser à La Revue de l'Utilisateur de l'IBM PC
2, rue d'Amsterdam
75009 Paris
Tél. : 47.64.07.57. - 42.67.93.80

Nom ..... **Prénom** .....
Adresse .....
.....
Code Postal ..... **Ville** .....
Téléphone .....
Veuillez m'envoyer ..... **logiciel (s) ATELIER au prix unitaire de 450 F (TTC).**
Veuillez trouver ci-joint un chèque de ..... **F en règlement à Softin Communications.**



Le dopage d'une machine ancienne n'est pas une mince affaire : nombreux sont les pièges à déjouer et les astuces à connaître.

Daniel VASSEUR



Bricolinfo

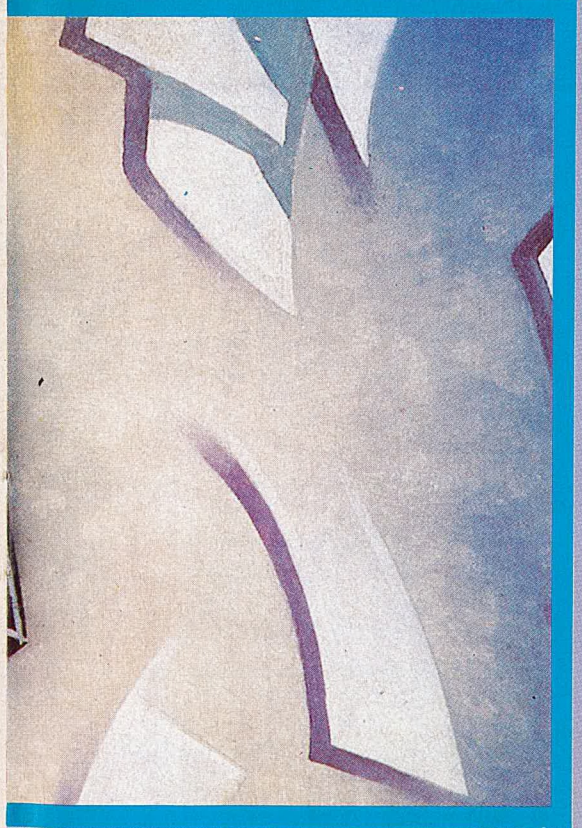
Le contexte

N'étant pas propriétaire d'une machine 386, je pourrai me sentir complexé par rapport aux performances affichées par les formules 1 de la micro-informatique moderne.

Et pourtant, aux commandes de mon AT/286, j'obtiens des vitesses de croisière qui laissent loin derrière nombre de machines 386.

Tout est venu d'un problème hardware rencontré sur mon bon vieil AT à 10 MHz qui, après 4 années d'un usage intensif, montrait des signes de fatigue évidents.

La carte d'extension mémoire longtemps soupçonnée d'infidélités s'est avérée être parfaitement saine, de même que le disque dur qui présentait des symptômes bizarres de dysfonctionnement.



En réalité, mais je ne l'ai su que bien plus tard, tout provenait du boîtier d'alimentation qui accusait le coup de ses milliers d'heures de fonctionnement.

A l'époque de son achat, mon AT était particulièrement performant : 10 MHz de vitesse d'horloge, 1 MO RAM sur carte-mère (répartition 512 + 512), c'était quelque chose.

Une seule gêne à signaler sur toute cette période : la réinitialisation du clavier ne se faisait pas toujours bien sur boot à chaud comme sur démarrage à froid. Pour retrouver les caractères spéciaux accessibles à l'aide de la touche ALTGR (CTRL-ALT-xx à l'époque), il fallait parfois relancer la machine plusieurs fois de suite.

Notez que ce problème existe aussi sur des compatibles de grande marque comme HP.

Parlons-en de la marque, je ne la connais pas car il n'y en pas ! Achetée chez un ensemblier qui pratique des prix "discount" — et chez qui on n'est jamais très sûr d'avoir du matériel neuf —, les différents éléments constituant la machine provenaient d'horizons divers, surtout là où se lève le soleil d'ailleurs.

Cette disparité d'approvisionnements m'a valu de ne jamais avoir la documentation relative à la carte multifonctions supportant supportant un port parallèle, un port joystick et deux ports séries.

Avec le temps, les logiciels évoluaient : au fil des nouvelles versions, les produits que j'utilise le plus nécessitaient de plus en plus de ressources machines, et certaines incompatibilités commençaient à montrer le bout du nez.

C'est ainsi que se sont révélés les problèmes d'affichage vidéo qui, renseignements pris, ne venaient pas de la carte compatible HERCULES, mais bien d'un problème de synchronisation sur le BUS.

Restons honnêtes ! La machine a fonctionné 3 bonnes années avant de montrer ses premières faiblesses, et vu le prix payé à l'époque c'est-à-dire environ 20 000 F alors que l'IBM AT II coûtait dans les 70 000 F à configuration équivalente, je suis gagnant autant sur le plan financier que sur celui des performances, puisqu'aujourd'hui, ces deux machines sont "dépassées".

Confronté au problème du remplacement total ou partiel à mener de front avec un budget ridicule, j'ai donc commencé à chercher des solutions.

Le marché actuel

La presse informatique qui se veut bien informée et au goût du jour, ne considère plus que les machines 386.

En dehors de 386-33, point de salut et il ne reste qu'une solution : passer à la caisse car, tous les "benchs" sont là pour le dire, pour se maintenir dans la spirale des performances, il faut une machine moderne et donc chère.

Au risque de passer pour un ringard, je ne suis toujours pas convaincu et je réfute cet argument souvent avancé par des commerciaux qui confondent mémoire conventionnelle et étendue tout aussi bien que connexion parallèle et série (ne souriez pas, je me suis trouvé plusieurs fois à expliquer la différence chez des "grands" de la distribution).

Pour propulser la vente de machines haut de gamme, les constructeurs proposent des machines construites autour du 386SX d'abord à 16 puis maintenant à 20 MHz, et les 286 disparaissent des catalogues.

Pourtant 386 ne veut pas dire performances, et dans la plupart des cas, vous travaillerez moins vite avec une machine 386SX plutôt qu'avec un 286 musclé.

A mon avis, un seul argument est recevable pour préférer un 386SX-16 à un 286-16 : la possibilité de faire tourner des logiciels écrits spécialement pour le 386 tel AUTOCAD 386. Mais dans ce cas, l'absence de performances est tellement sensible, que vous regretterez de n'avoir fait que la moitié de l'achat.

A la date d'aujourd'hui, le marché se tourne irrésistiblement vers WINDOWS 3 qui donne d'excellents résultats sur des machines 286 bien configurées.

Effectivement, vous n'atteindrez pas le mode "386 Etendu" qui est une façon d'obtenir le multitâche et la gestion de mémoire virtuelle, mais est-ce bien de cela dont vous avez besoin ?

L'homme a beaucoup de mal à s'adapter à une machine multitâche, et par expérience — je travaille souvent avec VM/386 — je sais que l'utilisateur trouve ses limites intellectuelles à 2 voire 3 tâches, sachant que je ne considère pas l'activité d'un spooler comme une tâche à part entière.

Ce dont l'homme a réellement besoin, c'est d'un "switcher" qui lui permette de ne pas quitter son traitement de texte pour passer à son tableur, et en cela, le mode "standard" de WINDOWS 3 est très efficace.

Les investissements en machine 386 avec assez de RAM ou de vitesse disque pour que nos nerfs supportent le "swap" du mode "386 Etendu", sont disproportionnés dans la plupart des cas.

Fort de ce raisonnement que je partage puisqu'il est mien, je me suis mis en quête d'une solution alliant mon besoin de performances à mes impératifs budgétaires et logiciels car je pense rester longtemps encore sous DOS.

Le voyage aux enfers

Ayant arrêté mon choix sur le principe d'une machine de type 286-16, j'ai entamé mes recherches pour trouver une carte-mère susceptible de remplacer ma vieille bombarde.

En région parisienne, les revendeurs pullullent, de même que les incompétents !

J'avais déjà eu affaire à des revendeurs capables d'installer des contrôleurs de disque pour XT sur des AT sans en informer leurs clients, mais j'étais loin de me douter de ce qui m'attendait au fond de l'abîme.

J'ai dû visiter une vingtaine de magasins qui, pour la plupart, se vendent en offrant des prix imbattables en complément du service indispensable que tout client est en droit d'attendre. A plusieurs reprises, je me suis senti infiniment petit devant l'immensité de l'ignorance des différents vendeurs et vendeuses à qui j'ai eu affaire. Je ne citerai que 3 exemples.

1) Parlant d'une carte d'interface parallèle à 150 F, l'un d'entre eux m'a affirmé qu'elle disposait de suffisamment d'intelligence pour se placer automatiquement sur une adresse de port disponible. Du jamais vu !

2) J'ai rencontré des cartes-mères qui "devaient d'abord être saturées en mémoire avant de savoir gérer d'éventuelles cartes d'extensions" (j'avais mon INTEL ABOVE BOARD à recycler).

3) Plus fort encore, du côté de la Bastille, une carte manifestement très fatiguée et d'une conception aussi ancienne que la forteresse défunte, m'a été présentée comme tournant à 12 MHz alors que le microprocesseur INTEL était certifié à 10 et que les indications de vitesse normalement portées sur le quartz avaient été malencontreusement effacées.
Et bien sûr, pas de possibilité de test préalable ni même d'échange en cas de problème. Pour décision immédiate, je bénéficiais d'un rabais exceptionnel de 150 F, ce qui aurait amené la facture à 1 650 F, ceci peut-être pour excuser les différentes soudures sauvages présentes un peu partout ainsi que l'aspect vraiment peu engageant du connecteur clavier.

Bref, j'ose le dire, j'éprouve un profond sentiment de honte pour une profession qui a beaucoup de mal à se débarrasser de ses bêtes noires. En plus dans l'opération, non content d'avoir largement contribué à la richesse de la ville de Paris par parcmètres interposés, j'avais perdu énormément de temps !

A l'horizon et alors que je commençais à désespérer, j'ai pourtant vu poindre la lueur du soleil levant dans un minuscule magasin du Boulevard Ménilmontant.

Dans un capharnaüm indescriptible où l'on trouve aussi bien un chien que des copies de menus de restaurants chinois, j'ai trouvé un matériel asiatique de bon aspect, des prix aussi intéressants que TTC, et, incroyable mais vrai, de réelles compétences !

Le matériel en question, je l'avais déjà rencontré dans plusieurs magasins, mais à des prix de 30 à 50 % plus élevés.

Une ombre au tableau : la qualité de l'accueil est discutable du fait du très faible effectif de cette entreprise et des délais d'attente qui résultent d'un décrochage systématique des 2 téléphones à chaque nouvelle sonnerie. Pourtant le sourire y est toujours présent de même qu'une tasse de café si vous avez le bon goût d'arriver au bon moment.

Dans ce micro-magasin j'ai donc trouvé la carte-mère que je cherchais :

— un 286 certifié à 16 MHz par AMD ;

— un BIOS AMI avec SETUP et diagnostic intégré ;

— jusqu'à 4 MO RAM sur carte-mère ;

— 6 slots dont 4 sur 16 bits ;

— une documentation très complète bien qu'en anglais.

Comble du bonheur, la carte est garantie 1 an même en cas d'implantation de ma CHARGECARD, alors que, sur le même matériel, d'autres me proposaient 3 mois ou rien du tout.

Après avoir demandé l'installation d'1 MO de RAM et le test (gratuit) de la carte, il me restait plus qu'à signer un chèque de 1 853 F.

Les affres du montage

De retour chez moi, j'ai entrepris le démontage-remontage de ma machine, et là, première surprise : les points de fixation de la carte-mère réputés normalisés, ne s'adaptaient que partiellement à mon ancien coffret.

Nouveau déplacement chez mon vendeur pour me procurer un boîtier moderne avec 4 baies en façade (3 au format 5"1/4 et 1 au format 3"1/2). Un boîtier équipé d'une alimentation 200 W ne coûte que 800 F, et toute la visserie et connectique est fournie.

A un prix aussi bas l'ensemble est d'une finition approximative, mais je lui pardonne ces petits défauts, y compris l'absence de documentation qui m'a obligé, au montage, à de longues hésitations et suppositions sur les branchements possibles des différents voyants installés sur le panneau frontal.

C'est dans ces moments là que l'on s'aperçoit que ce qui peut sembler la logique même n'est pas toujours la réalité.

Au bout de 3 heures — de quoi faire sourire tout monteur chevronné — j'avais terminé le remontage complet de la machine, ainsi que son paramètrage.

En configurant tout cela, j'ai découvert que le BIOS AMI supportait les 45 disques durs les plus courants. Pour mon disque SEAGATE ST251-1 de 40 MO qui se déclare désormais comme étant de type 40, je n'ai plus besoin du driver en assembleur qui me mangeait un peu de place en mémoire vive (voir l'article "Mariage DOS-PROLOGUE" dans l'exemplaire n° 67, Décembre 1990, de la Revue de l'Utilisateur de l'IBM PC).
De plus, pour tout type de disque non prévu, il est possible d'utiliser l'entrée 47 de la table pour une définition sur mesure des paramètres d'exploitation.

Cette nouvelle fonctionnalité est particulièrement appréciée depuis l'avènement des disques en interface AT BUS qui, bien sûr, ne sont pas prévus dans les tables des machines anciennes.

J'ai eu également confirmation que la carte-mère, bien qu'extensible à 4 MO RAM, ne supportait que les répartitions 1 MO par 8 composants DIP, et 2 ou 4 MO par 2 ou 4 barrettes SIMMS et à condition d'ôter les 8 chips que j'avais acquis. Mais là ce n'était pas une surprise car le vendeur, honnête, m'en avait informé.

Je pouvais enfin éprouver ma nouvelle bête de course.

Les nouvelles performances

Au niveau compatibilité, je n'ai eu aucune surprise. Seule l'interface clavier refuse mon utilitaire SPEEDKBD (voir l'exemplaire n° 52, Avril 1989, de la Revue de l'Utilisateur de l'IBM PC, "Clavier intelligent sur AT et PS/2), ce qui, bien que désagréable, n'est pas pénalisant en soi.

Dans la foulée, j'ai lancé QAPLUS de DIAGSOFT, utilitaire particulièrement pratique pour les opérations de maintenance et la détection des problèmes.

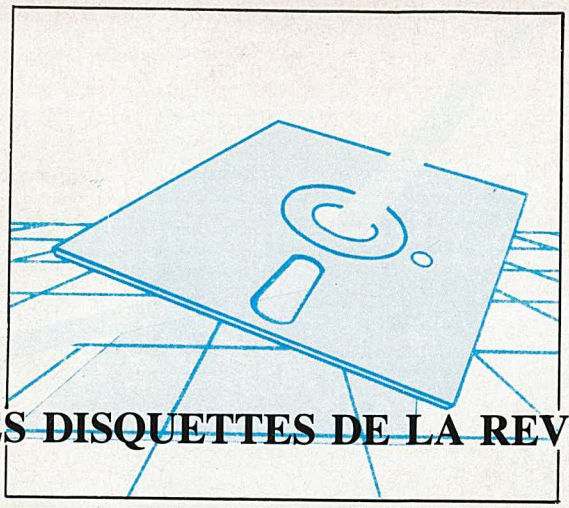
Au chapitre SYSINFO, j'ai obtenu des résultats édifiants :

— 3 793 dhrystones en mode turbo à 16 MHz, plus fort que le PS/2 8580 d'IBM, une machine 386DX à 16 MHz ;

— 1 979 dhrystones à 8 MHz, c'est-à-dire l'équivalent d'un PS/2 8550 nanti d'une horloge à 10 MHz.

Pour faire référence à un standard plus connu, je dirai que NORTON me renvoit un indice de performance de 17.6 en mode turbo et de 9.1 en mode standard.

Cela s'explique non seulement par la fréquence du quartz que QAPLUS m'a d'ailleurs mesurée à 15,68 MHz et non à 16 comme annoncée par le construc-



LES DISQUETTES DE LA REVUE

DISQUETTES THÉMATIQUES		
N°	Revues	
1	10,11	Simulateur du fonct. d'un micro
2	21 à 25	Debug système et assembleur
3	17	Séquentiel indexé en BASIC
4	19 à 23	Éditeur et DAO en BASIC
5	24 à 27	Étude TURBO PASCAL (jeux, math.)
6	28	Conversion fichier DOS/WORDSTAR
7	30, 32	Étude C et PROLOG
8	30	Jeu SMILEY
9	43, 47, 49	PRSC, XPRINT et param. imprim.
10	29, 30, 33, 34	Imp. & graph., stylescript.
11	26-28, 30-37, 38	Adresses utiles + bib. souris
12	52, 53	Clavier intelligent et PASSWORD
13	40, 49	Scrolling texte + MENUDER
14	50, 51	Fenêtres Pascal + masques saisie
15	32, 34, 40	Hâchurage, Scrolling HGR, XFEN
16	48, 54	Courbes fractales
17	39, 41, 42, 43	Courbes et volumes en 3D
18	34, 35	MINITEL 1
19	36, 37	MINITEL 2
20	40, 42	MINITEL 3
21	50, 52, 53	RT/CMOS, NEWAUX, INT 44H
22	46, 48, 49	MODIF/DEPL, SAUF, BATCH inter.
23	38, 39, 41	Résidents : ZIP5, TABLES, CONTEXT
24	42, 43, 44	Assemb. & DOS (ASM, HEURE, PRSC...)
25	52, 53	Gestion de fichier en TP4
26	40, 45, 54	DEDICORE, SCAN & DEL, SAUVE
27	44, 45, 40	Étude TP. 4.0 + Fenêtrage HGR
28	48 à 54	Dossier TURBO C
29	50, 54	SECUDOS et prog. invisible
30	51, 52, 53, 55	DAO en TURBO PASCAL
31	55, 56	DBASE
32	56, 58, 59, 60...	GENECR : Générateur d'écran en T.P.
33	58, 60	Compil. interp. et désas. PASCAL

UTILITAIRES		
N°	Revues	Titres
1	40	DEDICORE : Tri sélectif des répertoires pour copier, effacer, renommer
2	41	CONTEXT : Un résident pour connaître votre configuration hard & soft
3	43	PRSC : Un résident pour rediriger et paramétrer la recopie d'écran
4	45	SCAN & DEL : Pour sélectionner et effacer vos fichiers inutiles
5	47	XPRINT : Un programme résident pour configurer l'imprimante
6	49	PARAMIMP : Un résident pour contrôler et utiliser votre imprimante
7	54	LS : Une commande DIR style UNIX
8	57	VDISK : Un disque virtuel dynamique
9	57	COPYDIR : Pour copier vos répertoires sur disquettes
10	56	RESIZE : Optimisez la taille de l'environnement de votre PC
11	58	SOSFILE : Si vous voulez récupérer des fichiers endommagés, cet utilitaire est pour vous
12	58	PCTEST : Pour tout connaître sur la configuration de votre machine
13	56, 57, 59	EXFILE : Analyseur de fichiers PASCAL
14	58, 60	HARDCOPY : Impression graphique d'écran EGA-VGA en N. & B. et nuances de gris
15	61	EDITPAL : Éditeur de palette pour EGA/VGA
16	60	XMEM : Utilisation de la mémoire étendue sous DOS
17	64	ANTIVIRUS : Traitement des virus PingPong et Jérusalem (sources et programmes)

BON DE COMMANDE :
à adresser à SOFTIN COMMUNICATIONS
2 rue d'Amsterdam
75009 Paris - Tél. : 42.81.54.27

Nom .....

Adresse .....

Code Postal ..... **Ville** .....

Téléphone .....

Veuillez m'envoyer la ou les disquette (s) numéros : THÉMATIQUES (160,00 F) : .....
UTILITAIRES (90,00 F) : .....

Veuillez trouver ci-joint un chèque de..... **× 160,00 F soit**..... **Francs.**
..... **× 90,00 F soit**..... **Francs.**
Total :..... **Francs.**

en règlement à l'ordre de Softin Communications.

Date : **Signature :**

teur, mais aussi par le fabricant du microprocesseur, en l'occurrence AMD, qui a l'habitude de faire plus et mieux pour un prix moindre.

La cohérence de la carte-mère repose également sur le temps d'accès à la mémoire qui doit être d'environ 80ns si l'on veut éviter l'introduction d'états d'attente au niveau du microprocesseur.
Pour connaître les temps de réaction de la mémoire, j'ai utilisé MANIFEST de QUATERDECK qui m'a indiqué que le mode turbo accède la RAM 14.4 fois plus vite qu'un XT, contre 6.3 fois seulement en vitesse lente.

Bien entendu, le changement de carte-mère n'a rien modifié du comportement et des temps d'accès du disque dur. L'optimisation avait déjà été faite du temps de l'ancienne carte par utilisation de SPINRITE II de DISK MANAGER (cf La Revue de l'Utilisateur de l'IBM PC n° 69, Mars-Avril 1991), et là aussi, les temps de transferts sur le BUS et la capacité de réaction de la carte contrôleur sont des paramètres incompressibles.
A noter que certains 386 évolués (RS25 d'HP, STEP 386 d'EVEREX,...) ont des vitesses de BUS modifiables qui peuvent d'ailleurs parfois générer des problèmes de fonctionnement avec des cartes d'extensions non prévues pour réagir aussi rapidement.

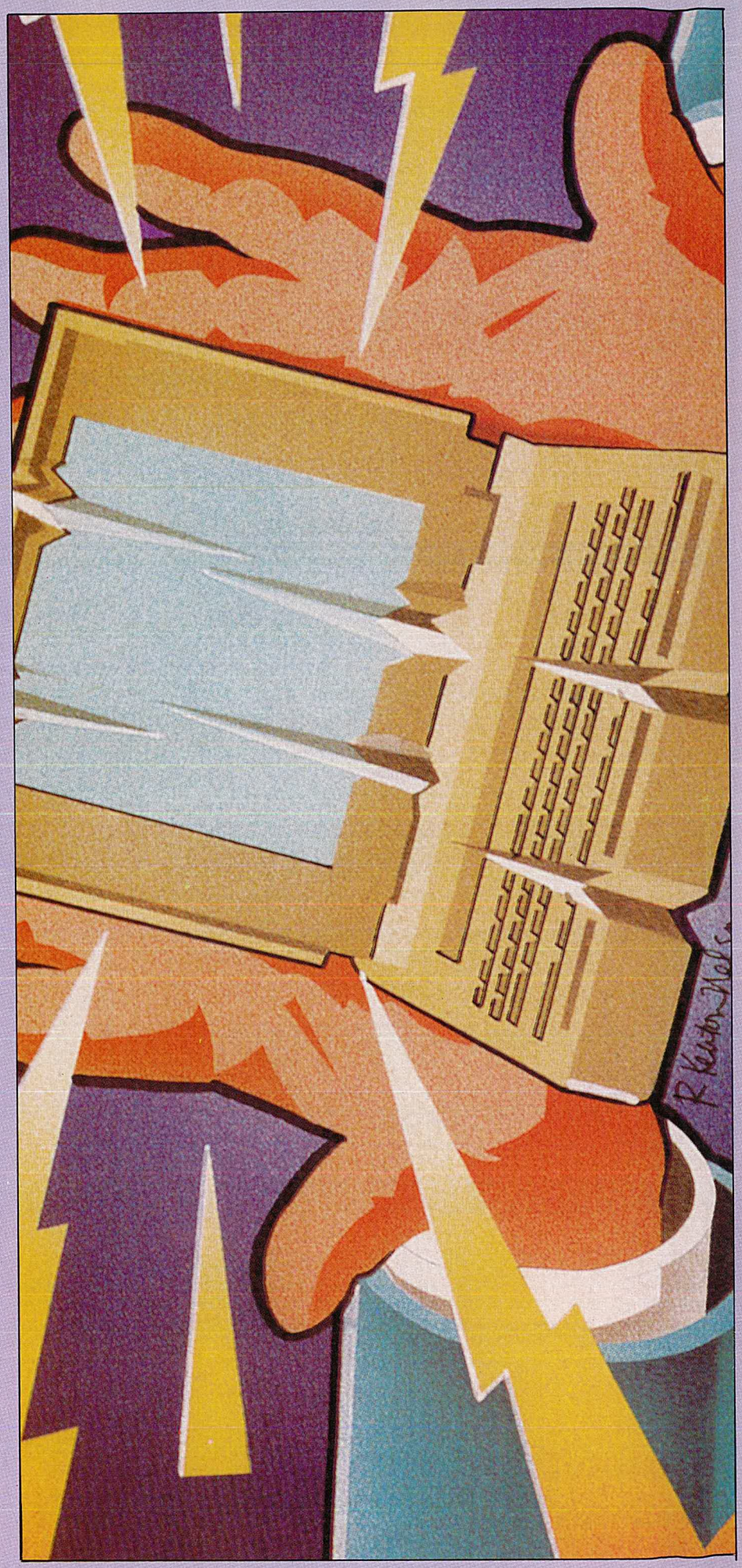
A vrai dire, je n'ai pas besoin du bench de QAPLUS pour constater que ma machine va vraiment très très vite.
A l'évidence, elle fonctionne toujours en mode turbo ce qui me permet d'avoir des affichages en mode texte quasiment instantanés (environ 13 500 caractères par seconde avec le turbo contre 8 500 en temps normal) y compris en mode graphique comme sous WINDOWS 3, alors que je n'ai toujours que ma bonne vieille carte bimode HERCULES/CGA dotée de composants RAM à 150ns de temps d'accès.

Le must

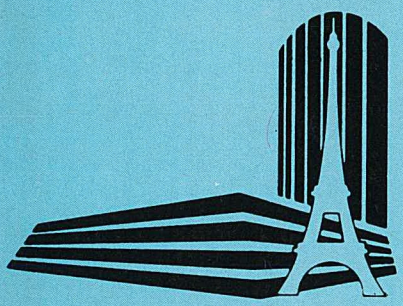
Ma nouvelle carte-mère possède une "lacune" : elle répartit automatiquement la mémoire disponible en 640 KO de conventionnelle et 384 KO d'étendue, ce qui, de nos jours, est parfaitement normal.

Les 2 MO de RAM disponibles sur ma carte d'extension ABOVE BOARD d'INTEL n'étaient donc plus accessibles qu'en mode EMS puisque cette carte ne sait installer de la mémoire étendue que sur une adresse frontière multiple de 512 KO.

Qui plus est, ma carte INTEL n'est pas hardwarement au niveau 4.0 de l'EMS : elle m'interdisait donc une configuration performante pour DESQview (voir à ce sujet l'article "EMS 4.0" dans la Revue de l'Utilisateur de l'IBM PC n° 69, Mars-Avril 1991).



11e Congrès de l'E.A.O. et du Multimédia



Les 7, 8 et 9 Janvier 1992
Palais des Congrès
Porte Maillot Paris 17e

POUR RETENIR UN STAND

Le Journal de la Formation Continue et de l'E.A.O.
2 rue d'Amsterdam
75009 Paris
Tél. : 47.64.07.57. - 42.67.93.80.

D'autre part, il m'aurait fallu attendre quelques semaines pour avoir la prochaine version de la carte-mère, celle qui serait apte à gérer la mémoire étendue EMS 4.0.
Sachant que les cartes d'extension supportant hardwarement l'EMS 4.0 sont aussi rares que chères, et qu'il vaut mieux être prudent car les vendeurs ne font pas la différence — les annonces publicitaires parlent toujours de compatibilité EMS 4.0 mais sans jamais préciser si cette compatibilité est matérielle ou logicielle — il me restait une solution : la vente d'occasion de ma carte INTEL m'a permis de racheter 4 MO de RAM en barrettes SIMMS tout en me remboursant le prix du méga déjà installé. Il suffisait ensuite de procéder à la réinstallation de ma ALL CHARGECARD.

FIGURE 1

sommateur de mémoire LIM ou EEMS, trouve son bonheur dans un mapping mémoire optimal que sur 286 la ALL CHARGECARD est seule à lui apporter.

Une telle configuration a des performances stupéfiantes qui n'ont rien à envier à celles des machines réputées plus puissantes, témoin les benchs précités.

A titre professionnel, je dispose d'une machine 386SX-16 de marque avec 4 MO de RAM. Au niveau vitesse, cette machine est très nettement inférieure à mon clone asiatique au prix de revient incomparable.

La périphérie

Qui plus est, la génération 386 d'entrée de gamme ne se distingue en rien de son ascendance 286 pour ce qui est

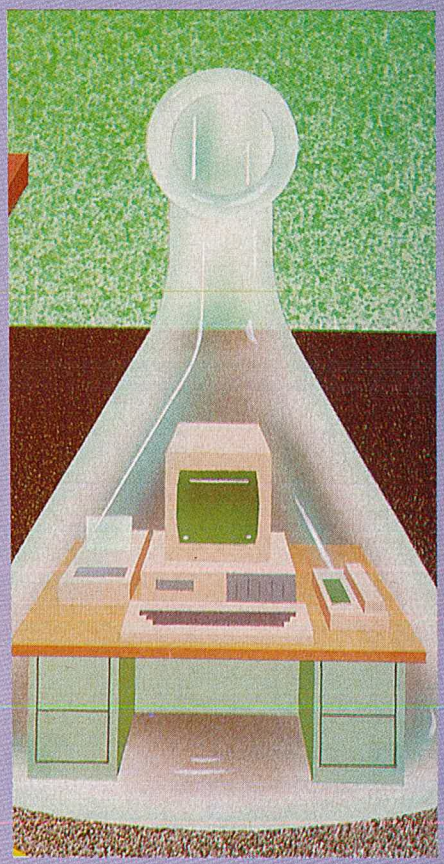
FIGURE N° 1
PERFORMANCES VIDEO

	EVEREX 8 bits		COMPAQ 16 bits	
Vitesse CPU	8 MHz	16 MHz	8 MHz	16 MHz
Sans Shadow ROM VGA	2 491	2 705	3 156	3 381
Avec Shadow ROM VGA	6 312	11 384	8 607	15 779

Tests réalisés en mode texte et exprimés en caractères par seconde
(QAPLUS de DIAGSOFT, version 4.21)

dées selon la technique RLL qui permet des temps d'accès inférieurs et des capacités de stockage améliorées à surface égale.

Là où la différence peut aussi se faire nettement sentir, c'est sur les accès mémoire. A l'ère des logiciels gourmands de mémoire étendue, l'extension de la mémoire sur carte-mère revêt un aspect de plus en plus important.



Ce dispositif hardware m'a non seulement permis de paginer une partie de ma mémoire étendue, mais m'a également donné 224 KO d'espace mémoire supplémentaire pour DOS, plus qu'assez pour toutes les couches bas niveau du système d'exploitation ainsi que pour mes TSR favoris.

En réservant 2 MO de mémoire étendue pour un RAMDRIVE et un peu de mémoire XMS, et en paginant le reste, je dispose alors d'une configuration fort pratique qui satisfait à la fois DOS, DESQview et WINDOW 3.

Au chapitre des gâteries, je me suis permis quelques tests avec une carte VGA 8 bits de chez EVEREX (chipset MAXLOGIC et RAM à 80ns) et une 16 bits de chez COMPAQ (chipset PARADISE et RAM à 120ns).

En activant les possibilités de "ROM shadowing" de la CHARGECARD sur le BIOS vidéo, QAPLUS m'a rapporté des indices de performances qui se passent de commentaires (cf figure n° 1).

Dans l'article "EMS 4.0", je vous ai montré comme DESQview, gros con-

d'une utilisation sous DOS, sauf peut-être sur le plan de la satisfaction intellectuelle. Il faut vraiment taper dans les très grosses machines pour voir apparaître des différences significatives.

En effet, les constructeurs dotent leurs yearlings d'une périphérie à la hauteur, et c'est bien par le haut de gamme que les disques réellement rapides (moins de 20ms) ont fait leur apparition. Mais aujourd'hui, un disque de 40 MO en interface AT BUS avec moins de 20ms de temps d'accès ne coûte guère plus cher qu'un bon vieux ST251-1 en interface ST506 : alors pourquoi en priverait-on les machines 286, pour autant qu'elles en valent la peine ?

Qu'est-ce que l'interface AT BUS ? Tout simplement le moyen de faire transiter les informations en provenance ou à destination du disque à travers une carte d'I/O 16 bits qui ne fait rien de plus que de servir de connecteur entre le BUS de la machine et le contrôleur installé sur disque lui-même.
Sur le disque, les données sont enco-

COMPAQ est l'un des premiers constructeurs à l'avoir ressenti car depuis plusieurs années déjà une partie de sa gamme supporte une grande quantité de mémoire sans rajout de carte sur le BUS (le modèle 386-25 par exemple utilise un slot particulier). Mais à quel prix

Quant à ses autres machines, elles restent désespérément limitées à 1 voire 2 MO sur carte-mère, ce qui, sous WINDOWS 3, permet seulement de commencer à travailler.

Or, pour un meilleur rendement, la mémoire ne devrait pas se trouver sur carte d'extension car, quelque soit la vitesse de réaction des composants, le temps d'accès est inexorablement limité par la vitesse et le taux de transfert du BUS.

Ceci est tellement vrai, que sur une machine très rapide de type 286-16, WINDOWS 3 plante en mode standard car incapable de gérer une disparité de temps d'accès sur deux plages de mémoire étendue, l'une sur carte-mère et l'autre sur carte d'extension.

Moralité, rien ne sert de mettre de la

COMMUNICATION
et
MULTIMÉDIA INTERACTIFS 92
Les 7, 8 et 9 Janvier 1992
Palais des Congrès
Paris (Porte Maillot)

APPEL AUX COMMUNICATIONS

Communication et Multimédia Interactifs 92 aura lieu les 7, 8 et 9 Janvier 1992 au Palais des Congrès (Porte Maillot, Paris 17e) dans le cadre du XIe Congrès de l'E.A.O. et du Multimédia.

Communication et Multimédia Interactifs qui s'est tenu en 1991 a, le premier, souligné l'importance du Multimédia. Cette manifestation qui a réuni les leaders du marché, IBM, BULL, INTEL, COMMODORE, PHILIPS et la participation de APPLE et MICROSOFT devient le rendez-vous annuel de la Communication et du Multimédia Interactifs.

De la même manière que les PC ont révolutionné la micro-informatique dans les années 1980, le Multimédia Interactif, qui consiste en la combinaison des PC et de l'ensemble de son environnement, des CD-ROM, du son de très haute fidélité, des images animées, du texte, le tout piloté par un hypermédia permettant de naviguer à travers d'immenses bases de données sera selon la prédiction de Bill GATES fondateur de MICROSOFT et créateur du MS-DOS "le plus important des marchés de la micro-informatique des années 1990".

Toutes les prévisions à l'horizon 1993, convergent vers un marché mondial du Multimédia Interactif de l'ordre de 65 Milliards de Francs.

Les systèmes multimédias interactifs modifieront dans les deux ans à venir, tous les outils de la communication, de la publicité, de la formation...

Le Comité de Programme s'intéresse aux communications qui proposent une réflexion concrète sur les besoins et les réponses apportés par la Communication et le Multimédia Interactifs permettant de faire bénéficier chaque participant des expériences menées et des conclusions tirées, mettant en évidence les spécificités et les qualités des outils du multimédia interactif, des méthodologies de création des multimédias interactifs et les applications de la Communication et du Multimédia Interactif.

La recherche des contributions porte sur les points suivants :

1°) PREMIÈRE JOURNÉE : LES OUTILS DE LA COMMUNICATION ET DU MULTIMÉDIA INTERACTIF :

— quelles configurations des micro-ordinateurs pour la Communication et le Multimédia Interactifs ?
— quelles technologies pour la Communication et le Multimédia Interactifs ?
— les hypertextes, les hypermédias,
— RNIS et Télématique,
— CD-I, CD-ROM, CD-ROM XA,
— son, images et base de données d'images,...

2°) DEUXIÈME JOURNÉE : MÉTHODOLOGIES DE CRÉATION DE LA COMMUNICATION ET DU MULTIMÉDIA INTERACTIFS.

— cahier des charges, d'un projet de Communication et Multimédia Interactifs,
— coûts des Communications et Multimédias Interactifs,
— validation des Communications Multimédias Interactifs,...

3°) TROISIÈME JOURNÉE : APPLICATIONS DE LA COMMUNICATION OU DU MULTIMÉDIA INTERACTIFS ET ÉVOLUTIONS PRÉVISIBLES.

— nouveaux besoins de Communication et du Multimédia Interactifs,
— applications de la Communication et du Multimédia Interactifs,
— Centres de Ressources des Communications et Multimédias Interactifs,
— les projets de Communication et de Multimédia Interactifs,
— les applications de Communication et du Multimédia Interactifs grand public,
— futur du marché du Multimédia Interactif,...
— utilisation du Multimédia Interactif,
— évolution des différents marchés du Multimédia Interactif (publicité, point de vente...).

En conséquence, les communications devront éviter les considérations théoriques sur les techniques de la Communication et du Multimédia Interactifs mais chercher plutôt la description des solutions et de leur contexte d'utilisation, les raisons des choix de l'application, l'intégration dans un système d'information ou de formation existant, la réorganisation de la distribution des tâches, l'impact commercial, l'impact stratégique et la rentabilité financière...

Cette liste n'est pas exhaustive et nous restons ouverts à toutes proposition pouvant intéresser les participants.

Les contributions proposées doivent parvenir au Comité du Programme impérativement avant le 20 Septembre 1991 et devront s'attacher à décrire en 2 pages leurs aspects pratiques accompagné d'un résumé de 15 lignes.

COUPON à adresser au Journal de la Formation Continue et de l'E.A.O.
2, rue d'Amsterdam, 75009 Paris
Tél. : 47.64.07.57. - 42.67.93.80

Nom ..... **Prénom** .....
Société .....
Adresse .....
Code Postal ..... **Ville** ..... **Pays** ..... **Tél** .....

souhaite :

☐ **Adresser pour la Communication et le Multimédia Interactifs 92 une communication sur le thème :** .....
..... **N°** .....
☐ **Souhaite recevoir le dossier-exposant pour envisager la location d'un stand.**
☐ **Souhaite recevoir le programme à sa parution (vers le 10 Novembre 1991).**

mémoire rapide sur des cartes d'extension classiques, la vitesse du BUS assurant la liaison avec le reste de la machine rendant tout luxe inutile. A ce niveau, pas la peine de faire riche car des circuits à 80ns n'apporteront rien de plus que des boîtiers à 120ns.

Par contre, ce qu'il faut à tout prix éviter, ce sont les circuits moins rapides comme les 150ns qui équipaient les premières cartes d'extension mémoire pour AT qui, à l'époque, tournaient sagement à 6 MHz.

Outre des difficultés de recyclage de périphériques anciens, la mise à niveau d'une machine peut cacher des petits pièges à l'origine de pertes de temps considérables.

Un problème fréquemment rencontré est celui des connecteurs qui ont changé de norme, bien que pour des raisons d'homogénéité les grands constructeurs aient conservé le standard précédent.

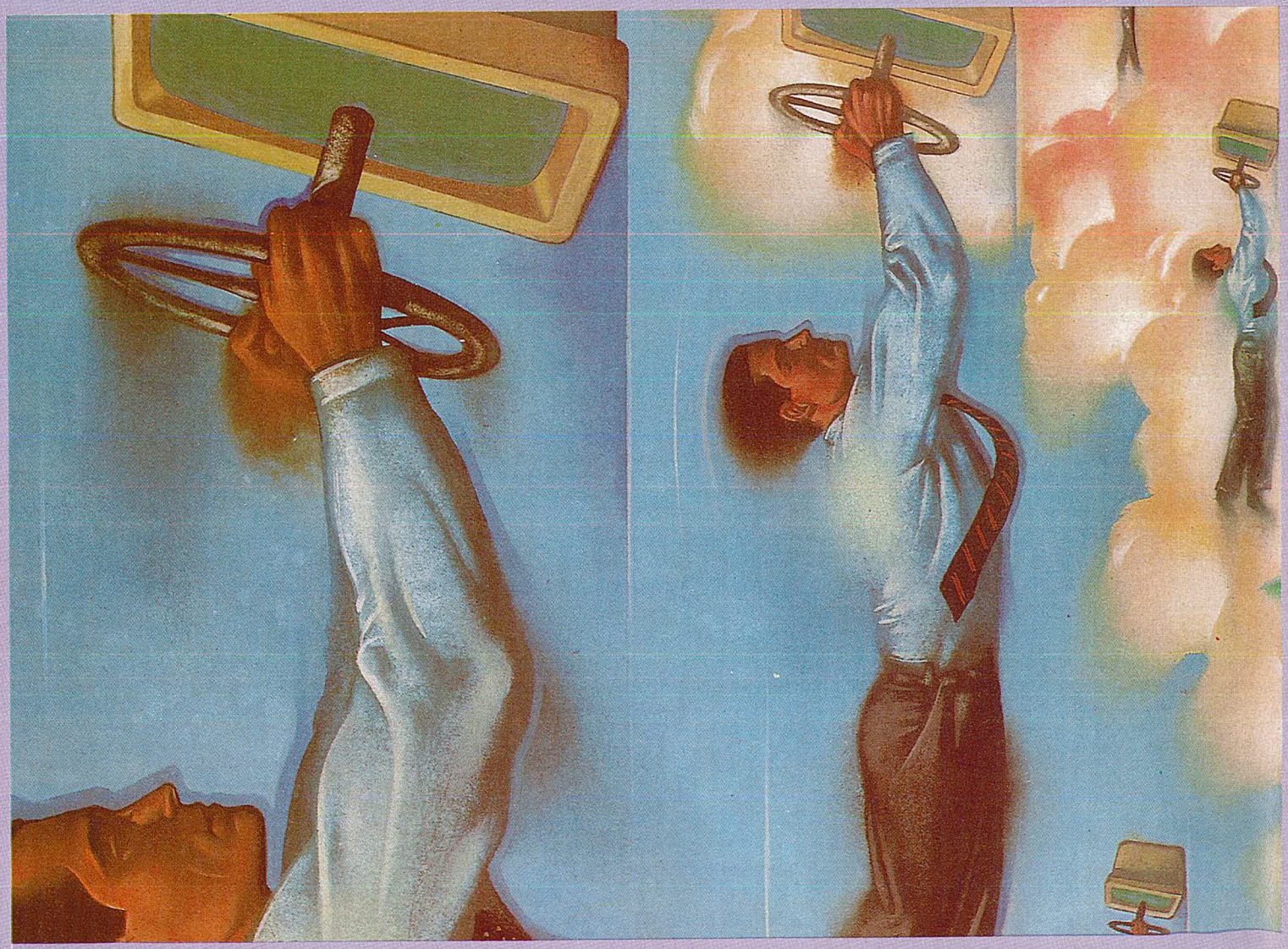
C'est le cas des lecteurs de disquette 3'' 1/2 qui, lorsqu'ils sont vendus pour des contrôleurs AT BUS, disposent d'un mini-connecteur d'alimentation ainsi que d'un nouveau type de branchement sur le câble des donnéees.

Chez mon vendeur de clones, le problème ne se pose pas : tout est prévu pour supporter toutes les connexions connues à ce jour, tant sur les câbles en nappe pour floppy que sur les sorties du boîtier d'alimentation.

Conclusion

Voilà comment avec 1 253 F j'ai pu transfigurer totalement ma machine tout en conservant la presque totalité de sa périphérie. Moyennant un investissement complémentaire de 800 F, j'ai remplacé la caisse de l'unité centrale et troqué mon alimentation 165 W usagée contre une 200 W toute neuve.

Une solution à base de carte-mère 386SX aux performances inférieures m'aurait coûté 1 600 F de plus chez ce même vendeur.



Tél. : 46.03.38.32
Télex : 250752 F
Télécopie : 46.05.28.71

LE MAP

LE MAP
5, Avenue du Maréchal Juin
92100 BOULOGNE

LOGICIEL ET MAINTENANCE D'APPLICATIONS

ECRANS

— VGA 14'' monochrome	835 F HT (990 F TTC)
— VGA couleur 14''	2 190 F HT (2 597 F TTC)
— 21'' couleur 1 280 × 1 024 CAO	11 990 F HT (14 220 F TTC)

IMPRIMANTES

— EPSON LQ 200	2 390 F HT (2 835 F TTC)
— STAR LC 20	1 390 F HT (1 649 F TTC)
— BULL 4/24 (136 col., 24 aig.)	4 590 F HT (5 444 F TTC)
— BULL LASER 6 p/mn - 1 Mo RAM	8 423 F HT (9 990 F TTC)
— CANON BJ 330	4 890 F HT (5 800 F TTC)

SAUVEGARDES

— ARCHIVE 60 MO FT 60	4 990 F HT (5 918 F TTC)

CARTES NOVELL/ETHERNET (10 M bits)

— NE 1000 8 bits bus AT	1 790 F HT (2 123 F TTC)
— NE 2000 16 bits bus AT	2 090 F HT (2 479 F TTC)
— NE 3000 32 bits bus EISA	6 090 F HT (7 223 F TTC)

DISPONIBLE

— CLAVIERS CYRILLIQUES
— TERMINAUX CYRILLIQUES
— CONSOMMABLES
— CABLES

Microsoft
MS-DOS
5.0

PRIX de la MAJ
666,10 F HT
(790 F TTC)

SPECIAL REMISE ⟩ 40 %

— Souris compatible	135 F HT (160 F TTC)
— Souris MICROSOFT US SERIE	790 F HT (937 F TTC)

LOGICIELS

— WINDOWS 3.0	1 190 F HT (1 411 F TTC)
— WORD pour WINDOWS	2 990 F HT (3 546 F TTC)
— EXCELL pour WINDOWS	2 990 F HT (3 546 F TTC)
— WORD PERFECT 5.1	2 990 F HT (3 546 F TTC)

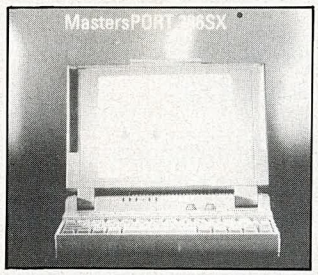
Revendeur agréé ZENITH data system Groupe BULL

REMISE EXCEPTIONNEL 37 %

PORTABLES ZENITH

— MINISPORT HD-4 : 80 C 88 10 MHz
— 1 MO RAM - DD 20 MO
— FD 3''1/2 - 2,5 Kg

PRIX LE MAP : 8 900 F HT (10 555 F TTC)



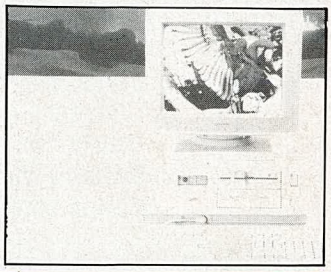
— MASTERS PORT 386 SX :
— 80386 SX - 20 MHz - 2 MO RAM
— VGA 32 niveaux - FD 3''1/2
— Disque dur 60 MO - 3 Kg

PRIX LE MAP : 17 990 F HT (21 336 F TTC)
+ CRAYON SOURIS MS : OFFERT

Revendeur agréé
DAEWOO
PLUS PRES DE VOUS

— 80286 / 12 MHz - 1 Mo RAM - VGA - 2,5
— 1 // - FD 3''1/2 - MS DOS 4.01

	ECRAN MONO	ECRAN COULEUR
DD 40 MO	6 790 F HT (8 053 F TTC)	7 990 F HT (9 476)
DD 80 MO	7 990 F HT (9 476 F TTC)	8 990 F HT (10 626 F TTC)



— 80386 SX - 16 MHz - 2 MO RAM - VGA
— 2 S - 1 // - FD 3''1/2
— MS DOS 4.01

	ECRAN MONO	ECRAN COULEUR
DD 40 MO	8 790 F HT (10 425 F TTC)	9 900 F HT (11 741 TTC)
DD 80 MO	9 900 F HT (11 741 F TTC)	10 900 F HT (12 927 F TTC)
DD 120 MO	10 900 F HT (12 927 F TTC)	12 900 F HT (15 299 F TTC)

La qualité de niveau ''Zéro-Défaut'', est le prix de la compétitivité de LE MAP

LE MAP

Revendeur agrée ZENITH data systems Groupe Bull

Revendeur agréé DAEWOO PLUS PRES DE VOUS

LA GAMME "PEARL" D' ALT COMPUTER

PS - 2010
- Processeur 8088 à 10 MHz
- 640 Ko de mémoire
- 5 slots d'extension
- 1 port série, 1 port parallèle et 1 port jeu
- Contrôleur disque et disquette
- 1 lecteur 360 Ko
- Clavier 102 touches
- Système d'exploitation DR-DOS
- Coffret SLIM. Alimentation 150 W

PS - 3012 (PS - 3016)
- Processeur 80286 à 12 MHz (16 MHz)
- 1 Mo de mémoire, extensible à 4 Mo
- 7 slots d'extension
- 2 ports série, 1 port parallèle et 1 port jeu
- Contrôleur disque et disquette AT-IDE
- 1 lecteur 1.2 Mo ou 1.44 Mo
- Clavier 102 touches
- Système d'exploitation DR-DOS
- Coffret BABY. Alimentation 200 W

PS - 4012 (PS - 4016)
- Identiques aux PS-3012 (PS-3016) avec coffret DESK-TOP

PS - 6016 (PS - 6020)
- Processeur 80386SX à 16 MHz (20 MHz)
- 1 Mo de mémoire, extensible à 8 Mo
- 7 slots d'extension
- 2 ports série, 1 port parallèle et 1 port jeu
- Contrôleur disque et disquette AT-IDE
- 1 lecteur 1.2 Mo ou 1.44 Mo
- Clavier 102 touches
- Système d'exploitation DR-DOS
- Coffret Mini-Tower. Alimentation 200 W

RECHERCHONS

DISTRIBUTEURS

et

REVENDEURS

ALT COMPUTER se réserve le droit de modifier, sans préavis, les caractéristiques de ses produits.

PS - 7025
- Processeur 80386 à 25 MHz
- 1 Mo de mémoire, extensible à 12 Mo
- 7 slots d'extension
- 2 ports série, 1 port parallèle et 1 port jeu
- Contrôleur disque et disquette AT-IDE
- 1 lecteur 1.2 Mo (5.25")
- Clavier 102 touches
- Système d'exploitation DR-DOS
- Coffret Tower. Alimentation 220 W

PS - 70C25 (PS - 70C33)
- Identique au PS-7025 avec 64 Mo de mémoire cache; vitesse du processeur 25 ou 33 MHz

PS - 80C25 (PS - 80C33)
- Processeur 80486 à 25 MHz (33 MHz)
- 1 Mo de mémoire, extensible à 16 Mo
- 128 Ko de mémoire cache
- 7 slots d'extension
- 2 ports série, 1 port parallèle et 1 port jeu
- Contrôleur disque et disquette AT-IDE
- 1 lecteur 1.2 Mo (5.25")
- Clavier 102 touches
- Système d'exploitation DR-DOS
- Coffret Tower. Alimentation 220 W

SCS - 3000
- Portable "Note Book"
- Processeur 80386SX à 20 MHz
- 1 Mo de mémoire, extensible à 4 Mo
- Ecran LCD VGA (480x640)
- 2 ports série, 1 port parallèle
- 1 disque dur 20 Mo
- 1 lecteur 1.44 Mo (3.5")
- Clavier 80 touches
- Batterie rechargeable
- Système d'exploitation DR-DOS
- Poids 3.5 Kg

Ces configurations sont livrées sur demande avec :

- Ecran 14" monochrome ou VGA couleur 1024 x 768 - Pitch = 0.28
- Disques durs de 40 Mo à 1 Go; contrôleurs AT-IDE, SCSI, ESDI.

PS - 4016

Processeur 80286 à 16 MHz

PS - 2010

Processeur 8088 à 10 MHz

PS - 3012

Processeur 80286 à 12 MHz

SCS - 3000

Processeur 80386SX à 20 MHz
ECRAN VGA (3.5 Kg)

PS - 6016

Processeur 80386SX
à 16 ou 20 MHz

PS - 70C25

Processeur 80386
à 25 ou 33 MHz

PS - 80C25

Processeur 80486
à 25 ou 33 MHz

ALT COMPUTER

1, Avenue de Laponie - Z.A. de Courtabœuf - 91951 LES ULIS CEDEX FRANCE
Tél. : 33 (1) 64.46.68.63 - Télex : 603 261 F ALTCHNO - Téléfax : 64.46.66.50

LE DS/3 ! LA MEILLEURE OFFRE DU MARCHÉ FRANÇAIS

AYEZ PITIÉ DE VOTRE POCHE ! MAIS D'ABORD DE VOS YEUX !

DYNAMIT COMPUTER

6 990 F TTC (5 893,76 F HT)
Si vous avez pitié de votre poche !
(avec Moniteur XYZ pitch 0,31)

7 690 F TTC (6 483,40 F HT)
Si vous avez pitié de vos yeux !
(avec Moniteur SONY TRINITRON pitch 0,25)

VOTRE PC 286/40 Mo COMPLET

Boîtier compact, alimentation 200W UL/FCC, carte mère 80286-12, indice Norton 15,8 1 Mo Ram, 1 Lecteur 3"1/2 1,44 Mo, 1 Disque dur 40 Mo Maxtor 24 ms. CARTES. vga 16Bits, carte contrôleur Lecteurs (2), Disque Dur (2), Série (2), Parallèle (1), Clavier 102 T, Dr Dos 5.0.

+ Garantie sur site 1 an Par TASQ (Filiale du CRÉDIT LYONNAIS !)

+ 1 Souris Neos Compatible Microsoft Made in Japan

+ SIDEKICK BORLAND

+ 1 Traitement de texte EN FRANÇAIS ! "DYNATEXT"

+ Communication Minitel MENTEL PC par ENTER

LA PREMIÈRE IMPRIMANTE LASER A 6 950,00 F TTC !

OKI 400 L : GARANTIE SUR SITE UN AN !

4 pages/minute, 512 Ko, idéal traitement de texte

PRIX RECORD OFFRE COMPLÈTE !

Imprimante portable BJ10e Jet d'encre 360 x 360 dpi
+ Bac feuille à feuille + Cable

Prix public (4 130,00 F HT)	4 898,00 F TTC
Prix DYNAMIT (2 862,56 F HT)	3 395,00 F TTC

CITIZEN SWIFT 24	2 950,00 F TTC
STAR LC20	1 990,00 F TTC
STAR LC200 COULEUR	2 990,00 F TTC
CANON BJ10e - Jet d'encre	2 790,00 F TTC

CONSULTEZ-NOUS POUR LA GAMME STAR, OKI

LECTEURS DE DISQUETTES

5" 1/4 1,2 Mo	490,00 F TTC
3" 1/2 1,44 Mo	490,00 F TTC
Berceau 5" 1/4 ♦ 3" 1/2	70,00 F TTC

DISQUES DURS

12 Mo BASF MFM	695,00 F TTC
4 Mo MAXTOR ATBUS	1 950,00 F TTC
80 Mo MAXTOR ATBUS	3 990,00 F TTC
110 Mo NEC ATBUS	4 790,00 F TTC
STREAMER 40/60 Mo	2 590,00 F TTC

MONITEURS

VGA MONOCHROME	950,00 F TTC
VGA COULEUR P. 0,31	2 490,00 F TTC
VGA SONY P. 0,25	3 690,00 F TTC
VGA MULTISCAN SONY	5 250,00 F TTC
NEC 2A	4 290,00 F TTC
NEC 3D	5 290,00 F TTC
NEC 4D	10 250,00 F TTC
NEC 5D	20 900,00 F TTC

TOUS LES LOGICIELS MS-DOS A PRIX CASSÉS, EXEMPLES :

	PRIX OFFICIEL TTC	PRIX DYNAMIT TTC	PRIX ÉDUCATION TTC
WORD 5.5 :	5 325,00 F	3 565,00 F	1 890,00 F
WORD WINDOWS	5 918,00 F	3 965,00 F	1 990,00 F
EXCEL	5 918,00 F	3 490,00 F	1 990,00 F

• Compétence Réseau Novell : nous installons et formons sur toute la France. Devis à la demande • Compétences Télématiques : installation de serveur minitel sous UNIX • Les cartes mères de nos ordinateurs 386-sx16/20, 386-40, 486-25 sont fabriquées aux USA. Nos ordinateurs équipent de nombreuses banques ! Écrivez-nous pour un catalogue complet de nos produits !

Erreurs et omissions exceptées.

Je désire avoir plus d'information sur le produit suivant :

Nom : _____
Prénom : _____
Téléphone : _____
Profession : _____
Adresse : _____

DYNAMIT COMPUTER
77, rue de Maubeuge - 75010 PARIS
Tél. : 42 82 17 09 - Fax : 42 82 17 25
***Mardi - vendredi** 9 h 30 - 19 h – **Samedi** 10 h - 18 h*

MM2 COMMUNICATION - 45 94 95 95